

The microtype package

An interface to the micro-typographic extensions of pdf \TeX

R Schlicht
w.m.l@gmx.net

v2.3d
2009/03/27

Abstract

The `microtype` package provides a \LaTeX interface to the micro-typographic extensions of pdf \TeX : most prominently, character protrusion and font expansion, furthermore the adjustment of interword spacing and additional kerning, as well as hyphenatable letterspacing (tracking) and the possibility to disable all or selected ligatures. It allows to apply these features to customisable sets of fonts, and to configure all micro-typographic aspects of the fonts in a straight-forward and flexible way. Settings for various fonts are provided.¹

Note that font expansion and character protrusion will only work with pdf \TeX , at least version 0.14f. Automatic font expansion requires version 1.20 or newer. Disabling ligatures requires pdf \TeX 1.30, letterspacing and the adjustment of interword spacing and of kerning requires version 1.40. The package will by default enable protrusion and expansion if they can safely be assumed to work. These two features are also available with lua \TeX . The `microtype` package does not work with Xe \TeX .

The alternative package `letterspace`, which also works with plain \TeX , provides the user commands for letterspacing only, omitting support for all other extensions (see section 7).

This package is copyright © 2004–2009 R Schlicht. It may be distributed and/or modified under the conditions of the [\$\LaTeX\$ Project Public License](#), either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘author-maintained’.

¹ Currently, this package provides protrusion settings for Computer Modern Roman, Palatino, Times, URW Garamond, Adobe Garamond and Minion, Bitstream Charter and Letter Gothic, the AMS symbols and Euler fonts, for various Euro symbol fonts, as well as some generic settings for unknown fonts (cf. table 3 on page 21). Contributions are very welcome.

Contents

1	Micro-typography with pdf\TeX	4
2	Getting started	5
3	Options	6
3.1	Enabling the micro-typographic features	6
3.2	Character protrusion	7
3.3	Font expansion	8
3.4	Tracking/letterspacing	9
3.5	Miscellaneous options	9
3.6	Changing options later	10
4	Selecting fonts for micro-typography	10
5	Micro fine tuning	12
5.1	Character protrusion	13
5.2	Font expansion	14
5.3	Tracking	16
5.4	Interword spacing	18
5.5	Additional kerning	19
5.6	Character inheritance	20
5.7	Configuration files	20
6	Context-sensitive setup	22
7	Letterspacing revisited	24
8	Disabling ligatures	25
9	Hints and caveats	25
10	Contributions	28
11	Acknowledgments	28
12	References	29
13	Short history	30
14	Implementation	33
14.1	Preliminaries	34
	Debugging [35] Requirements [37] Declarations [40] Auxiliary macros [41] Compatibility [48]	
14.2	Font Setup	52
	Protrusion [57] Expansion [64] Interword spacing (glue) [66] Additional kerning [68] Tracking [69] Disabling ligatures [78] Loading the configuration [79] Translating characters into slots [83] Hook into \LaTeX 's font selection [88] Context-sensitive setup [91]	

14.3	Configuration	93
	Font sets [93] Variants and aliases [99] Disabling ligatures [100]	
	Interaction with babel [100] Fine tuning [101] Character inheritance [107]	
	Permutation [109]	
14.4	Package options	112
	Declaring the options [112] Reading the configuration file [116] Hook	
	for other packages [117] Changing options later [118] Processing the	
	options [120]	
15	Configuration files	128
15.1	Font sets	128
15.2	Font variants and aliases	129
15.3	Interaction with babel	130
15.4	Note on admissible characters	130
15.5	Character inheritance	131
	OT1 [131] T1 [131] LY1 [132] OT4 [132] QX [133] T5 [134] Euro	
	symbols [134]	
15.6	Tracking	135
15.7	Font expansion	135
15.8	Character protrusion	136
	Default [137] Italics [145] Small caps [153] Italic small caps [156] Text	
	companion [157] Computer Modern math [162] AMS symbols [166]	
	Euler [170] Euro symbols [174]	
15.9	Interword spacing	175
	Nonfrenchspacing [176]	
15.10	Additional kerning	178
	French [178] Turkish [179]	
16	Auxiliary file for micro fine tuning	179
A	Change history	182
B	Index	189
C	The L^AT_EX Project Public License	197

List of Tables

1	Availability of micro-typographic features	7
2	Predefined font sets	12
3	Fonts with tailored protrusion settings	21
4	Order for matching font attributes	81

1 Micro-typography with pdf_{TEX}

pdf_{TEX}, the _{TEX} extension written by Hàn Th   Thành, introduces a number of micro-typographic features that make it the tool of choice not only for the creation of electronic documents but also of works of outstanding time-honoured typography: most prominently, *character protrusion* (also known as margin kerning) and *font expansion*. Quoting Hàn Th   Thành’s thesis:

After you have read the text on the right, you can view the effect of the features it describes by clicking on the links:

Protrusion	off
Expansion	off

Both features are enabled throughout this document.

‘Margin kerning is the adjustments of the characters at the margins of a typeset text. A simplified employment of margin kerning is hanging punctuation. Margin kerning is needed for optical alignment of the margins of a typeset text, because mechanical justification of the margins makes them look rather ragged. Some characters can make a line appear shorter to the human eye than others. Shifting such characters by an appropriate amount into the margins would greatly improve the appearance of a typeset text.

Composing with font expansion is the method to use a wider or narrower variant of a font to make interword spacing more even. A font in a loose line can be substituted by a wider variant so the interword spaces are stretched by a smaller amount. Similarly, a font in a tight line can be replaced by a narrower variant to reduce the amount that the interword spaces are shrunk by. There is certainly a potential danger of font distortion when using such manipulations, thus they must be used with extreme care. The potentiality to adjust a line width by font expansion can be taken into consideration while a paragraph is being broken into lines, in order to choose better breakpoints.’ [Thành 2000, p. 323]

Both these features have been lacking a simple L_{ATEX} user interface for quite some time. Then, the `pdfcprot` package was released, which allowed L_{ATEX} users to employ character protrusion without having to mess much with the internals.

Font expansion, however, was still most difficult to utilise, since it required that the font metrics are available for all levels of expansion. Therefore, anybody who wanted to make use of this feature had to create multiple instances of the fonts in advance. Shell scripts to partly relieve the user from this burden were available – however, it remained a cumbersome task. Furthermore, all fonts were still being physically created, thus wasting compilation time and disk space.

In the summer of 2004, Hàn Th   Thành implemented a feature that has proven as a major facilitation for _{TEX} and L_{ATEX} users: font expansion can now take place automatically. That is, pdf_{TEX} no longer needs the expanded font metrics but will calculate them at run-time and completely in memory.

After this great leap in usability had been taken, the development did not stop. On the contrary, pdf_{TEX} was extended with even more features: version 1.30 introduced the possibility to *disable all ligatures*, version 1.40 a robust *letterspacing* command, the *adjustment of interword spacing* and the possibility to specify *additional character kerning*.

Robust and hyphenatable *letterspacing (tracking)* has always been extremely difficult to achieve in _{TEX}. Although the `soul` package undertook great efforts in making this possible, it could still fail in certain circumstances; even to adjust the tracking of a font throughout the document remained impossible. Employing pdf_{TEX}’s new extension, this no longer poses a problem. The `microtype` package

provides the possibility to change the tracking of customisable sets of fonts, e. g., all small capitals. It also introduces two new commands `\textls` and `\lststyle` for ad-hoc letterspacing, which can be used like the normal text commands. Note that letterspacing only works in PDF mode.

Adjustment of interword spacing is based upon the idea that in order to achieve a uniform greyness of the text, the space between words should also depend on the surrounding characters. For example, if a word ends with an ‘r’, the following space should be a tiny bit smaller than that following, say, an ‘m’. You can think of this concept as an extension to T_EX’s ‘space factors’. However, while space factors will influence all three parameters of interword space (or glue) by the same amount – the kerning, the maximum amount that the space may be stretched and the maximum amount that it may be shrunk – pdfT_EX provides the possibility to modify these parameters independently from one another. Furthermore, the values may be set differently for each font. And, probably most importantly, the parameters may not only be increased but also decreased. This feature may enhance the appearance of paragraphs even more. Emphasis in the last sentence is on the word ‘may’: this extension is still highly experimental – in particular, only ending characters will currently have an influence on the interword space. Also, the settings that are shipped with microtype are but a first approximation, and I would welcome corrections and improvements very much. I suggest reading the reasoning behind the settings in section 15.9.

Setting *additional kerning* for characters of a font is especially useful for languages whose typographical tradition requires certain characters to be separated by a space. For example, it is customary in French typography to add a small space before question mark, exclamation mark and semi-colon, and a bigger space before the colon and the guillemets. Until now, this could only be achieved by making these characters active (for example by the `babel` package), which may not always be a robust solution. In contrast to the standard kerning that is built into the fonts (which will of course apply as usual), this additional kerning is based on single characters, not on character pairs.

The possibility, finally, to *disable all ligatures* of a font may be useful for type-writer fonts.

The microtype package provides an interface to all these micro-typographic extensions. All micro-typographic aspects may be customised to your taste and needs in a straight-forward manner. The next chapters will present a survey of all options and customisation possibilities.

2 Getting started

There is nothing surprising in loading this package:

```
\usepackage{microtype}
```

This will be sufficient in most cases, and if you are not interested in fine-tuning the micro-typographic appearance of your document (which would seem unlikely, since using this package is proof of your interest in typographic issues), you may

actually skip the rest of this document. If this, on the other hand, does not satisfy you – be it for theoretical or practical reasons – this manual will guide you on the path to the desired results along the following milestones:

- Enable the respective micro-typographic feature, either via the respective package option or with the `\microtypesetup` command (section 3).
- Select the fonts to which this feature should be applied by declaring and activating ‘sets of fonts’. Some sets are predefined, which may be activated directly in the package options (section 4).
- Fine-tune the micro-typographic settings of the fonts or sets of fonts (section 5).
- If you’re of the kind who always wants to march on, you’ll certainly be interested in the possibility of context-sensitive setup (section 6).
- You are even countenanced to leave the path of typographic virtue and steal some sheep (section 7) or trespass in other ways (section 8).
- Should you encounter any obstacles, follow the hints and caveats (section 9).

3 Options

Like many other L^AT_EX packages, the `microtype` package accepts options in the well known key=value syntax. In the following, you’ll find a description of all **keys** and their possible values (‘true’ may be omitted; multiple values, where allowed, must be enclosed in braces; the default value is shown on the right, preceded by an asterisk if it is contingent on the pdf_TE_X version and/or the output mode).

3.1 Enabling the micro-typographic features

protrusion true, false, compatibility, nocompatibility, ** *true

expansion These are the main options to control the level of micro-typographic refinement which the fonts in your document should gain. By default, the package is moderately greedy: character protrusion will be enabled, font expansion will only be disabled in circumstances where pdf_TE_X cannot expand the fonts automatically, that is, if it is either too old (versions before 1.20) or if the output mode is DVI (see section 3.5). In other words, `microtype` will try to apply as much micro-typography as can safely be expected to work under the respective conditions (and it is usually not necessary to load the package with different options for PDF resp. DVI mode).

activate Protrusion and expansion may be enabled or disabled independently from each other by setting the respective key to true resp. false. The `activate` option is a shortcut for setting both options at the same time. Therefore, the following lines all have the same effect (when creating PDF files with a recent version of pdf_TE_X):

```
\usepackage[protrusion=true,expansion]{microtype}
```

```
\usepackage[activate={true,nocompatibility}]{microtype}
```

```
\usepackage{microtype}
```

Table 1: Availability of micro-typographic features

T _E X engine			Micro-typographic features					
Engine	Version	Output	Protrusion	Expansion	(= auto)	Kerning	Spacing	Tracking
pdfT _E X	< 0.14f	DVI/PDF	∅	∅	∅	∅	∅	∅
	≥ 0.14f	DVI/PDF	★	☒	∅	∅	∅	∅
	≥ 1.20	DVI	★	☒	∅	∅	∅	∅
		PDF	★	★	★	∅	∅	∅
	≥ 1.40	DVI	★	☒	∅	☒	☒	∅
		PDF	★	★	★	☒	☒	☒ ^a
luaT _E X	≥ 0.25	DVI	★	☒	∅	∅	∅	∅
		PDF	★	★	★	∅	∅	∅
★ = enabled			☒ = not enabled		∅ = not available		a ≥ 1.40.4 recommended	

When pdfT_EX employs font expansion and character protrusion, line breaks (and consequently, page breaks) may turn out differently. If this is not desired – because you are re-typesetting a book whose pagination must not change – you may pass the value `compatibility` to the protrusion and/or expansion options. Typographically, however, the results will be suboptimal, hence the default value is `nocompatibility`.

Finally, you may also specify the name of a font set to which character protrusion and/or font expansion should be restricted. See section 4 for a detailed discussion. Specifying a font set for a feature implicitly activates this feature.

tracking

true, false, **

false

spacing

There is no compatibility level for the new extensions of tracking, interword spacing, and additional kerning. Therefore, they can only be switched on or off, or they may be activated by passing a set name to the option. By default, neither feature is enabled.

In table 1, you find an overview of which micro-typographic features are available and enabled by default for the relevant pdfT_EX versions and output modes.

Whether ligatures should be disabled cannot be controlled via a package option but by using the `\DisableLigatures` command, which is explained in section 8.

3.2 Character protrusion

factor

<integer>

1000

Using this option, you can globally increase or decrease the amount by which the characters will be protruded. While a value of 1000 means that the full protrusion as specified in the configuration (see section 5.1) will be used, a value of 500 would result in halving all protrusion factors of the configuration. This might be useful if you are generally satisfied with the settings but prefer the margin kerning to be less or more visible (e. g., if you are so proud of being able to use this feature that you want everybody to see it, or – to mention a motivation more in compliance

with typographical correctness – if you are using a large font that calls for more modest protrusion).

unit character, $\langle dimension \rangle$ character

This option is described in section 5.1, apropos the command `\SetProtrusion`. Use with care.

3.3 Font expansion

auto true, false * true

As noted in chapter 1, the expanded versions of the fonts may be calculated automatically. This option is true by default provided that pdfTeX's version is found to be 1.20 or higher and the output mode is PDF; otherwise, it will be disabled. If auto is set to false, the fonts for all expansion steps must exist (with files called $\langle font\ name \rangle \pm \langle expansion\ value \rangle$, e. g., `cmr12+10`, as described in the [pdfTeX manual](#)).

Automatic font expansion does not work with bitmap fonts. Therefore, if you are using the Computer Modern Roman fonts in T1 encoding², you should either install the cm-super fonts or use the Latin Modern fonts (package `lmodern`).

stretch $\langle integer \rangle$ 20

shrink You may specify the stretchability and shrinkability of a font, i. e., the maximum amount that a font may be stretched or shrunk. The numbers will be divided by 1000, so that a stretch limit of 10 means that the font may be expanded by up to 1%. The default stretch limit is 20. The shrink limit will by default be the same as the stretch limit.

step $\langle integer \rangle$ * 1

Fonts are not expanded by arbitrary amounts but only by certain discrete steps within the expansion limits. With recent versions of pdfTeX (1.40 or newer), this option is by default set to 1, in order to allow pdfTeX to try the maximum number of font instances, and hence to guarantee the best possible output. One problem that may occur here is that pdfTeX runs out of memory with huge documents; in this case, read about the error messages in the hints and caveats section (9), or try with a larger step. Older pdfTeX versions, however, had to include every font instance in the PDF file, which may increase the file size quite dramatically. Therefore, in case you are using a pre-1.40 pdfTeX version, step is by default set to one fifth of the smaller value of stretch and shrink.

selected true, false false

When applying font expansion, it is possible to restrict the expansion of some characters that are more sensitive to deformation than others (e. g., the 'O', in contrast to the 'I'). This is called *selected expansion*, and its usage allows to increase the stretch and shrink limits (to, say, 30 instead of 20); however, the gain is limited since at the same time the average stretch variance will be decreased. Therefore, this option is by default set to false, so that all characters will be expanded by the same amount. See section 5.2 for a more detailed discussion.

² En passant, it may be noted that Type 1 format and T1 encoding are in no other way related than that both start with a 'T' and end with a '1'.

3.4 Tracking/letterspacing

letterspace *<integer>* 100

This option changes the default amount for tracking (see section 5.3) resp. letter-spacing (see section 7). The amount is specified in thousandths of 1 em; admissible values are in the range of -1000 to $+1000$.

3.5 Miscellaneous options

DVIoutput true, false * false

pdf \TeX is not only able to generate PDF output but can also spit out DVI files.³ The latter can be ordered with the option `DVIoutput`, which will set `\pdfoutput` to zero.

Note that this will confuse packages that depend on the value of `\pdfoutput` if they were loaded earlier, as they had been made believe that they were called to generate PDF output where they actually weren't. These packages are, among others: `graphics`, `color`, `hyperref`, `pstricks` and, obviously, `ifpdf`. Either load these packages after `microtype` or else issue the command `\pdfoutput=0` earlier – in the latter case, the `DVIoutput` option is redundant.

When generating DVI files, font expansion has to be enabled explicitly. Neither `letterspacing` nor *automatic* font expansion will work because the postprocessing drivers (`dvips`, `dvipdfm`, etc.) resp. the DVI viewer are not able to generate the fonts on the fly.

draft true, false false

final If the `draft` option is passed to the package, *all micro-typographic extensions will be disabled*, which may lead to different line, and hence page, breaks. The `draft` and `final` options may also be inherited from the class options; of course, you can override them in the package options. E. g., if you are using the class option `draft` to show any overfull boxes, you should load `microtype` with the `final` option.

verbose true, false, errors, silent false

Information on the settings used for each font will be written into the log file if you enable the `verbose` option. When `microtype` encounters a problem that is not fatal (e. g., an unknown character in the settings, or non-existent settings), it will by default only issue a warning and try to continue. Loading the package with `verbose=errors` will turn all warnings into errors, so that you can be sure that no problem will go unnoticed. If on the other hand you have investigated all warnings and decide to ignore them, you may silence `microtype` with `verbose=silent`.

babel true, false false

Loading the package with the `babel` option will adjust the typesetting according to the respective selected language. Read section 6 for further information.

config *<file name>* microtype

Various settings for this package will be loaded from a main configuration file, by default `microtype.cfg` (see section 5.7). You can have a different configuration file loaded instead by specifying its name *without the extension*, e. g., `config=mycrottype`.

³ Recent \TeX systems are using pdf \TeX as the default engine even for DVI output.

3.6 Changing options later

`\microtypesetup` {<key = value list>}

Inside the preamble, this command accepts all package options described above (except for `config`). In the document body, this command may be used to change the general settings of the micro-typographic extensions. It then accepts all options from section 3.1: `expansion`, `protrusion` and `activate`, which in turn may receive the values `true`, `false`, `compatibility` or `nocompatibility`, and `tracking`, `spacing` and `kerning` with the admissible values `true` or `false`. Passing the name of a font set is not allowed. Using this command, you could for instance temporarily disable font expansion by saying:

```
\microtypesetup{expansion=false}
```

4 Selecting fonts for micro-typography

By default, character protrusion will be applied to all text fonts that are being used in the document, and a basic set of fonts will be subject to font expansion. You may want to customise which fonts should get the benefit of micro-typographic treatment. This can be achieved by declaring and activating ‘font sets’; these font sets are specified via font attributes that have to match.

`\DeclareMicrotypeSet` [*features*] {<set name>} {<set of fonts>}

`\DeclareMicrotypeSet*` This command declares a new set of fonts to which the micro-typographic extensions should be applied. The optional argument may contain a comma-separated list of features to which this set should be restricted. The starred version of the command declares *and* activates the font set at the same time.

The *set of fonts* is specified by assigning values to the NFSS font attributes: encoding, family, series, shape and size (cf. [L^AT_EX 2_ε font selection](#)). Let’s start with an example. This package defines a font set called ‘`basictext`’ in the main configuration file as follows:

```
\DeclareMicrotypeSet{basictext}
{ encoding = {OT1,T1,LY1,OT4,QX,T5},
  family   = {rm*,sf*},
  series    = {md*},
  size      = {normalsize,footnotesize,small,large}
}
```

If you now call

```
\UseMicrotypeSet[protrusion]{basictext}
```

in the document’s preamble, only fonts in the text encodings OT1, T1, LY1, OT4, QX or T5, roman or sans serif families, normal (or ‘medium’) series, and in sizes called by `\normalsize`, `\footnotesize`, `\small` or `\large`, will be protruded. Math fonts, on the other hand, will not, since they are in another encoding. Neither will fonts in bold face, or huge fonts. Etc.

If an attribute list is empty or missing – like the ‘shape’ attribute in the above example – it does not constitute a restriction. In other words, this is equivalent to specifying *all* possible values for that attribute. Therefore, the predefined set ‘alltext’, which is declared as:

```
\DeclareMicrotypeSet{alltext}
{ encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
```

is far less restrictive. The only condition here is that the encoding must match.

If a value is followed by an asterisk (like ‘rm*’ and ‘sf*’ in the first example), it does not designate an NFSS code, but will be translated into the document’s `\(value)default`, e. g., `\rmdefault`.⁴ A single asterisk means `\(attribute)default`, e. g., `\encodingdefault`, respectively `\normalsize` for the size axis. Sizes may either be specified as a dimension (‘10’ or ‘10pt’), or as a size selection command *without* the backslash. You may also specify ranges (e. g., ‘small-Large’); while the lower boundary is included in the range, the upper boundary is not. Thus, ‘12-16’ would match 12 pt, 13.5 pt and 15.999 pt, for example, but not 16 pt. You are allowed to omit the lower or upper bound (‘-10’, ‘large-’).

Additionally to this declaration scheme, you can add single fonts to a set using the ‘font’ key, which expects the concatenation of all font attributes, separated by forward slashes, i. e., ‘font = `<encoding>/<family>/<series>/<shape>/<size>`’. This allows you to add fonts to the set that are otherwise disjunct from it. For instance, if you wanted to have the roman family in all sizes protruded, but only the normal sized, possibly italic, typewriter font (in contrast to, say, the small one), this is how you could declare the set:

```
\DeclareMicrotypeSet[protrusion]
{ myset }
{ encoding = T1,
  family   = rm*,
  font      = {T1/tt*/m/n/*,
               T1/tt*/m/it/*} }
```

As you can tell from the example, the asterisk notation is also allowed for the font key. A single asterisk is equivalent to ‘*/*/*/*/*’, i. e., the normal font. Size selection commands are possible, too, however, ranges are not allowed.

Table 2 lists the nine predefined font sets. They may also be activated by passing their name to the feature options protrusion, expansion, tracking, spacing and kerning when loading the package, for example:

```
\usepackage[protrusion=allmath,tracking=smallcaps]{microtype}
```

`\UseMicrotypeSet` [`<features>`] {`<set name>`}

This command activates a font set previously declared by `\DeclareMicrotypeSet`. Using the optional argument, you can limit the application of the set to one or more features. This command only has an effect if the feature was activated in the package options.

⁴ These translations will take place `\AtBeginDocument`, which means that changes to the defaults inside the preamble will also be taken into account. Only in cases where you change font defaults `\AtBeginDocument` yourself, you need to load `microtype` after these changes.

Table 2: Predefined font sets

Set name	Font attributes				
	Encoding	Family	Series	Shape	Size
all	Ø	Ø	Ø	Ø	Ø
alltext (allmath)	Text encodings, TS1 (OML, OMS, U)	Ø	Ø	Ø	Ø
basictext (basicmath)	Text encodings (OML, OMS)	\rm*, \sf*	\md*	Ø	\normalsize, \footnotesize, \small, \large
smallcaps	Text encodings	Ø	Ø	sc	Ø
footnotesize	Text encodings, TS1	Ø	Ø	Ø	-\small
scriptsize	Text encodings, TS1	Ø	Ø	Ø	-\footnotesize
normalfont	\encoding*	\family*	\series*	\shape*	\normalsize
‘Text encodings’ = OT1, T1, LY1, OT4, QX, T5					‘...*’ = ‘...default’

`\DeclareMicrotypeSetDefault` [*features*] {*set name*}

If a feature is enabled but no font set has been chosen explicitly, the sets declared by this command will be activated. By default, the ‘alltext’ font set will be used for character protrusion and additional kerning, the ‘basictext’ set for font expansion and interword spacing, and the ‘smallcaps’ set for tracking.

These commands may only be used in the preamble or in the main configuration file. Their scope is global to the document. Only one set per feature may be activated.

5 Micro fine tuning

Every character asks for a particular protrusion, spacing or kerning amount. It may also be desirable to restrict the maximum expansion of certain characters. Furthermore, since every font looks different, settings have to be specific to a font or set of fonts. This package offers flexible and straight-forward methods of customising these finer aspects of micro-typography.

All fine-tuning commands follow basically the same syntax: they all take three arguments; the first one is optional and may contain additional options; in the second argument, you specify the set of fonts to which the settings should apply; the third argument contains the actual settings.

The set of fonts to which the settings should apply is declared using the same syntax of *font axis* = *value list* pairs as for the command `\DeclareMicrotypeSet` (see section 4). The only difference is that asterisked values will be translated immediately instead of at the end of the preamble. To find the matching settings for a given font the package will try all combinations of font encoding, family, series, shape and size, with decreasing significance in this order. For instance, if both settings for the current family (say, T1/cmr//) and settings for italic fonts in the normal weight (T1//m/it/) exist, those for the cmr family would apply.⁵ The encoding must always match.

⁵ For the interested, table 4 on page 81 presents the exact order.

5.1 Character protrusion

`\SetProtrusion` [*options*] {*set of fonts*} {*protrusion settings*}

Using this command, you can set the protrusion factors for each character of a font or a set of fonts. A very incomplete example would be the following:

```
\SetProtrusion
{ encoding = T1,
  family   = cmr }
{ A          = {50,50},
  \textquoteleft = {700, } }
```

which would result in the character ‘A’ being protruded by 5% of its width on both sides, and the left quote character by 70% of its width into the left margin. This would apply to all font shapes, series and sizes of the T1 encoded Computer Modern Roman family.

The *protrusion settings* consist of *<character>* = *<protrusion factors>* pairs.

The characters may be specified either as a single character (‘A’), as a text symbol command (‘\textquoteleft’), or as a slot number: three digits for decimal notation, prefixed with ‘#’ for hexadecimal, with ‘o’ for octal (e. g., the ‘fl’ ligature in T1 encoding: 029, #1D, o35). 8-bit (and even UTF-8) characters may be entered directly or in L^AT_EX’s traditional 7-bit notation: both ‘\’A and ‘A are valid, provided the character is actually declared in both the input and the font encoding. Note that you also have the possibility to declare lists of characters that should inherit settings (see section 5.6).

The protrusion factors designate the amount that a character should be protruded into the left margin (first value) respectively into the right margin (second value). By default, the values are relative to the character widths, so that a value of 1000 means that the character should be shifted fully into the margin, while, for example, with a value of 50 it would be protruded by 5% of its width. Negative values are admitted, as well as numbers larger than 1000 (but effectively not more than 1 em of the font). You can omit either number if the character should not be protruded on that side, but must not drop the separating comma.

Options:

name You may assign a name to the protrusion settings, so that you are able to load it by another list.

load You can load another list (provided, you previously assigned a name to it) before the current list will be loaded, so that the fonts will inherit the values from the loaded list.

Thus, the configuration may be simplified considerably. You can for instance create a default list for a font; settings for other shapes or series can then load these settings, and extend or overwrite them (since the value that comes last will take precedence). Font settings will be loaded recursively. The following options will affect all loaded lists:

factor This option can be used to influence all protrusion factors of the list, overriding any global factor setting (see section 3.2). For instance, if you want

fonts in larger sizes to be protruded less, you could load the normal lists, just with a different factor applied to them:

```
\SetProtrusion
[ factor = 700
  load   = cmr-T1 ]
{ encoding = T1,
  family   = cmr,
  size     = large- }
{ }
```

unit By default, the protrusion factors are relative to the respective character's width. The `unit` option may be used to override this and make `microtype` regard all values in the list as thousandths of the specified width. Issuing, for instance, `'unit=1em'` would have the effect that a value of, say, 50 now results in the character being protruded by 5% of an em of the font (thus simulating the internal measuring of pdfTeX's `\lcode` and `\rcode` primitives). The default behaviour can be restored with `unit=character`.⁶

preset Presets the protrusion codes of all characters to the specified values (`={\left},{right}`)), possibly scaled by a factor. A `unit` setting will only be taken into account if it is not `=character`.

inputenc Selects an input encoding that should apply to this list, regardless of what the document's input encoding is. You may specify any encoding that can be loaded via the `inputenc` package, e. g., `ansinew`, `koi8-r`, `utf8`.

context The scope of the list may be limited to a certain context. For an example application, see section 6.

5.2 Font expansion

\SetExpansion [*options*] {*set of fonts*} {*expansion settings*}

By default, all characters of a font are allowed to be stretched or shrunk by the same amount. However, it is also possible to limit the expansion of certain characters if they are more sensitive to deformation. This is the purpose of the `\SetExpansion` command. Note that it will only have an effect if the package was loaded with the selected option (cf. section 3.3). Otherwise, the expansion settings will be ignored – unlike the options in the optional first argument, which will still be evaluated.

If the package was loaded with the selected option, and settings for a font don't exist, font expansion will not be applied to this font at all. Should the extraordinary situation arise that you want to employ selected expansion in general but that all characters of a particular font (*set*) should be expanded or shrunk by the same amount, you would have to declare an empty list for these fonts.

The *expansion settings* consist of *<character> = <expansion factor>* pairs. You may specify one number for each character, which determines the amount that a character may be expanded. The numbers denominate thousandths of the full expansion.

⁶ The `unit` option can even be passed globally to the package (cf. section 3.2). However, all provided settings are created under the assumption that the values are relative to the character width. Therefore, you should only change it if you are certain that the default settings will not be used in your document.

For example, if you set the expansion factor for the character ‘O’ to 500, it will only be expanded or shrunk by one half of the amount that the rest of the characters will be expanded or shrunk. While the default value for character protrusion is 0 – that is, if you didn’t specify any characters, none would be protruded – the default value for expansion is 1000, which means that all characters would be expanded by the same amount.

Options:

name, load, preset, inputenc, context Analogous to `\SetProtrusion`, the optional argument may be used to assign a name to the list, to load another list, to preset all expansion factors, to set the input encoding, or to determine the context of the list (expansion contexts are only possible with pdfTeX version 1.40.4 or newer).

auto, stretch, shrink, step These keys can be used to override the global settings from the package options (see section 3.3). If you don’t specify either one of stretch, shrink and step, their respective global value will be used (that is, no calculation will take place).

As a practical example, suppose you have a paragraph containing a widow that could easily be avoided by shrinking the font a little bit more. In conjunction with the context option (see section 6 for further details), you could thus allow for more expansion in this particular paragraph:

```
\SetExpansion
[ context = sloppy,
  stretch = 30,
  shrink   = 60,
  step     = 5 ]
{ encoding = {OT1,T1,TS1} }
{ }
% ... END PREAMBLE
{\microtypecontext{expansion=sloppy}%
This paragraph contains an 'unnecessary' widow.}
```

This method of employing contexts to temporarily apply different expansion parameters only works with pdfTeX version 1.40.4 or later (for older versions, a dirty trick is laid out in section 14.2 on page 54). Also note that pdfTeX prohibits the use of fonts with different expansion limits or steps (even of different fonts) within one paragraph, hence the sloppy context has to be applied to complete paragraphs.

factor This option provides a different method to alter expansion settings for certain fonts, working around the restriction just mentioned. The factor option influences the expansion factors of all characters (in contrast to the overall stretchability) of the font. For instance, if you want the italic shape to be expanded less, you could declare:

```
\SetExpansion
[ factor = 500 ]
{ encoding = *,
  shape    = it }
{ }
```

The factor option can only be used to *decrease* the stretchability of the characters, that is, it may only receive values smaller than 1000. Also, it can only be used for single fonts or font sets; setting it globally in the package options wouldn't make much sense – to this end, you use the package's stretch and shrink options.

5.3 Tracking

`\SetTracking` [*options*] {*set of fonts*} {*tracking amount*}

An important typographic technique – which was missing in T_EX for a long time – is the adjustment of tracking, i. e., the uniform addition or subtraction of letter space to/from all the characters in a font. For example, it is good typographic practice to slightly space out text set in all capitals or small capitals (as in this document). Legibility may also be improved by minimally increasing the tracking of smaller and decreasing that of larger type.⁷ The `\SetTracking` command allows to specify the tracking amount for different fonts or font sets. It will also be evaluated by the `\textls` command, which may be used for letterspacing shorter pieces of text (see section 7).

The *tracking amount* is specified in thousandths of 1 em (or the given unit); negative values are allowed, too.

Options:

name, unit, context These options serve the same functions as in the previous configuration commands. The unit may be any dimension, default is 1 em.

spacing When the inter-*letter* spacing is altered, the inter-*word* spacing probably also needs to be adjusted. This option expects three numbers for interword space, stretch and shrink respectively, which are given in thousandths of 1 em (or of the current unit). If a value is followed by an asterisk, it denotes thousandths of the respective font dimension which will be added to it. For instance, with

```
\SetTracking[ spacing = {25*,166, } ] { encoding = *, shape = sc } { 25 }
```

the interword space will be increased by 2.5%, the stretch amount will be set to 0.166 em, while the shrink amount will be left untouched. If you don't specify the spacing option, the interword space will be scaled by the current letterspace amount (as in the above example), while stretch and shrink will not be changed.

outer spacing If an interword space immediately precedes or follows letter-spaced text, it will by default be equal to that within the text. With this option, which accepts the same values as spacing, it may be adjusted independently.

outer kerning If, on the other hand, no interword space precedes or follows, you may still want to slightly set off the first and last letter from adjoining letters. This option expects the kerning amounts for left and right hand side, separated by a comma, in thousandths of 1 em (or the current unit). If a value is followed by an asterisk, it denotes thousandths of the current letterspacing amount. A single asterisk means '500*'; this is also the default, i. e., the sum of the outer kerns is by

⁷ With full-featured fonts like Computer Modern, this is usually not necessary, though, since they come in optical sizes, and the tracking of the small-capitals font is already adjusted.

default equal to the current letterspace amount. To remove kerning on both sides, you would write ‘outer kerning={0,0}’.

no ligatures As far as pdf \TeX is concerned, ligatures in letterspaced fonts would be constructed as usual, which may be advisable when changing the tracking by only a small amount. For larger letterspacing amounts, on the other hand, the normal letter space within ligatures would have displeasing effects. This key expects a comma-separated list of characters for which ligatures should be disabled; only the character that begins a ligature must be specified. If the key is given without a value, *all* ligatures of the font will be disabled. This is not recommended, however, since it also entails that kerning will be switched off.⁸ The default settings disable ligatures for the character ‘f’ only, i.e., ‘ff’, ‘fi’, ‘ffi’, etc.⁹ In exceptional situations, you can manually break up a ligature by inserting ‘{\kern0pt}’ resp. babel’s “| shortcut, or protect it by enclosing it in \lslig (see section 7).

Since a picture is worth a thousand words, probably even more if, in our case, it depicts a couple of letterspaced words, let’s bring one to sum up these somewhat confusing options. Suppose you had the following settings (which I would in no way recommend; they are only for illustrative purposes):

```
\SetTracking
[ no ligatures = {f},
  spacing      = {600*,-100*, },
  outer spacing = {450,250,150},
  outer kerning = {*,*} ]
{ encoding = * }
{ 160 }
```

and then write:

```
Stop \textls{stealing sheep}!
```

this is the (typographically dubious) outcome:

Stop stealing sheep!

While the word ‘Stop’ is not letterspaced, the space between the letters in the other two words is expanded by the *tracking amount* of $160/1000\text{ em} = 0.16\text{ em}$. The *inner space* within the letterspaced text is increased by 60%, while its *stretch* amount is decreased by 10% and the *shrink* amount is left untouched. The *outer space* (of 0.45 em) immediately before the piece of text may *stretch* by 0.25 em and *shrink* by 0.15 em. Note that there is no outer space after the text, since the exclamation mark immediately follows; instead, the default *outer kern* of half the letterspace amount (0.08 em) is added. Furthermore, one *ligature* wasn’t broken up, because we neglected to specify the ‘s’ in the `no ligatures` key.

⁸ The inseparable connexion of ligatures and kerns is a limitation of \TeX that will not be lifted before the advent of lua \TeX .

⁹ With pdf \TeX versions older than 1.40.4, *all* ligatures, and hence all kerning, will be disabled. It is therefore recommended to use at least version 1.40.4.

Click on the image to show the kerns and spacings involved. Click on emphasised words in the text below to reveal the relation of image and code.

As another, more realistic example, suppose you want to space out all small capitals by 50/1000 em, fonts smaller than `\small` by 0.02 em, and to decrease the tracking of large type by 0.02 em. You can achieve this with the following settings:

```
\usepackage[tracking=true]{microtype}
\DeclareMicrotypeSet*[tracking]{my}
{ encoding = *,
  size      = {-small,Large-},
  font      = */*/*/sc/* }
\SetTracking[no ligatures = f]{ encoding = *, shape = sc}{ 50 }
\SetTracking{ encoding = *, size = -small }{ 20 }
\SetTracking{ encoding = *, size = Large- }{ -20 }
```

Letterspaced fonts for which settings don't exist will be spaced out by the default of 0.1 em (adjustable with the package option `letterspace`, see section 3.5). Suppose your editor wants you to shorten your 1000 pages chef-d'œuvre by a handful of pages, you could load `microtype` with (fingers crossed):

```
\usepackage[tracking=alltext,letterspace=-40]{microtype}
```

5.4 Interword spacing

`\SetExtraSpacing` [*options*] {*set of fonts*} {*spacing settings*}

This command allows you to fine tune the interword spacing (also known as glue). A preliminary remark on what a 'space' is may be in order: between two words, \TeX will insert a so called glue, which is characterised by three parameters – the normal distance between two words, the maximum amount of space that may be added to it, and the maximum amount that may be subtracted. The latter two parameters come into effect whenever \TeX tries to break a paragraph into lines and does not succeed; it can then stretch or shrink the spaces between words. These three parameters are specific to each font.

On top of these glue dimensions, \TeX has the concept of 'space factors'. They may be used to increase the space after certain characters, most prominently the punctuation characters. If pdf \TeX 's additional spacing adjustment is in effect, space factors are ignored, since it may be considered an extension to space factors with much finer control.

The *spacing settings* are declared as pairs of $\langle character \rangle = \langle spacing factors \rangle$, where the latter consist of three numbers: first, the additional kern inserted after this character if it appears before an interword space, second, the additional stretch amount, and third, the additional shrink amount. All values may also be negative, in which case the dimensions will be decreased. Not all values have to be specified, however, the settings must contain the two separating commas.

Options:

`name`, `load`, `factor`, `preset`, `inputenc`, `context` These options serve the same function as in the previous configuration commands.

unit You can specify the unit by which the specified numbers are measured. Possible values are: `character`, a *⟨dimension⟩* and, additionally, `space`. The latter will measure the values in thousandths of the respective space dimension set by the font. By default, the unit is measured by the space dimensions. For example, with these (nonsensical) settings:

```
\SetExtraSpacing
[ unit = space ] % default
{ font = */*/*/*/* }
{
  . = {1000,1000,1000},
}
```

the space inserted after a full stop would be doubled (technically speaking: $2 \times \text{\fontdimen 2}$), as would the maximum stretch and shrink amounts of the interword space (`\fontdimen 3` and `4`). Conversely, setting all three values to -1000 would completely cancel a space after the respective character.

5.5 Additional kerning

`\SetExtraKerning` [*⟨options⟩*] {*⟨set of fonts⟩*} {*⟨kerning settings⟩*}

With this command, you can fine tune the extra kerning. In contrast to standard kerning, which is always associated with a *pair* of characters, and to tracking, which specifies the space between *all* characters of a font, the extra kerning relates to single characters, that is, whenever a particular character appears in the text, the specified kerning will be inserted, regardless of which character precedes resp. follows it.

I should not neglect to mention a limitation of this additional kerning: words *immediately following* such a kern (not separated by a space) will not be hyphenated, unless you insert the breakpoints manually, e. g., for kerning after the apostrophe, ‘`l 'apos\ -trophe`’. This restriction of pdfTeX will hopefully be lifted soon.

The *kerning settings* are specified as pairs of *⟨character⟩ = ⟨kerning values⟩*, where the latter consist of two values: the kerning added before the character, and the kerning appended after the respective character. Once again, either value may be omitted, but not the separating comma.

Options:

name, load, factor, preset, inputenc These options serve the same function as in the previous configuration commands.

unit Admissible values are: `space`, `character` and a *⟨dimension⟩*. By default, the values denote thousandths of 1 em.

context When it comes to kerning settings, this option is especially useful, since it allows to apply settings depending on the current language.

For example, you can find the following settings, intended to be used for documents written in French, in the main configuration file:

```
\SetExtraKerning
[ name      = french-default,
  context   = french,
  unit      = space ]
{ encoding = {OT1,T1,LY1} }
{
  : = {1000,}, % = \fontdimen2
  ; = {500, }, % ~ \thinspace
  ! = {500, },
  ? = {500, }
}
```

What is the result of these settings? If they are active, like in the current paragraph, a thin space will be inserted in front of each question mark, exclamation mark and semicolon; a normal space in front of the colon. Read section 6 to learn how to activate these settings! This paragraph was input like this :

```
\begin{microtypecontext}{\kerning=french}
What is the result of these settings? If they are active, like in the
current paragraph, a thin space will be inserted in front of each
question mark, exclamation mark and semicolon; a normal space in front
of the colon. Read section~\ref{sec:context} to learn how to activate
these settings! This paragraph was input like this:
\end{microtypecontext}
```

5.6 Character inheritance

```
\DeclareCharacterInheritance [⟨features⟩] {⟨set of fonts⟩} {⟨inheritance lists⟩}
```

In most cases, accented characters should inherit the settings from the respective base character. For example, all of the characters À, Á, Â, Ã, Ä, Å and Æ should probably be protruded by the same (absolute) amount as the character A. Using the command `\DeclareCharacterInheritance`, you may declare such classes of characters, so that you then only have to set up the respective base character. With the optional argument, which may contain a comma-separated list of features, you can confine the scope of the list. Additionally, it accepts the `inputenc` key to set the input encoding for this list. The font set can be declared in the usual way, with the only exception that exactly one encoding must be specified. The inheritance lists are declared as pairs of *(base character) = (list of inheriting characters)*. Unless you are using a different encoding or a very peculiarly shaped font, there should be no need to change the default character inheritance settings.

In the main configuration file `microtype.cfg` and the other font-specific configuration files, you can find examples of all these commands.

5.7 Configuration files

The default configuration, consisting of inheritance settings, declarations of font sets and alias fonts, and generic protrusion, expansion, spacing and kerning settings, will

Table 3: Fonts with tailored protrusion settings

Font family (NFSS code)	Features	
	Encodings	Shapes
Generic	OT1, T1, LY1, QX, (TS1) ^a	n, (it, sl, sc) ^a
Computer Modern Roman (cmr) ^b	OT1, OT4, T1, T5, LY1, TS1	n, it, sl, sc
Bitstream Charter (bch) ^c	OT1, T1, T5, LY1, TS1	n, it, (sl) ^d , sc
Adobe Garamond (pad, padx, padj)	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc
URW Garamond (ugm) ^e	OT1, T1, TS1	n, it
Bitstream Letter Gothic (blg) ^f	OT1, T1, TS1	n, it
Adobe Minion (pmnx, pmnj) ^g	OT1, T1, LY1, TS1	n, it, (sl) ^d , sc, si
Palatino (ppl, pplx, pplj) ^h	OT1, OT4, T1, LY1, (TS1) ^a	n, it, (sl) ^d , sc
Times (ptm, ptmx, ptmj) ⁱ	OT1, OT4, T1, LY1, QX, (TS1) ^a	n, it, (sl) ^d , sc
Computer Modern math (cmsy, cmm)	OML/OMS	n/it
AMS symbols (msa, msb)	U	n
Euler (eur, eus, euf) ^j	U	n
Euro symbols (Adobe, ITC, marvosym)	U/OT1	n, it

^a Incomplete
^b Aliases: Latin Modern (lmr), ae (aer), zefonts (zer), eco (cmor), hfoldsty (hfor)
^c Aliases: mathdesign/Charter (mdbch), MicroPress's chmath (chr)
^d Settings inherited from italic shape
^e Alias: mathdesign/URW Garamond (mdugm)
^f Alias: ulgothic (ulg)
^g By courtesy of Harald Harders (h.harders@tu-bs.de)
^h Aliases: pxfonts (pxr), qfonts/QuasiPalatino, T_EX Gyre Pagella (qpl), FPL Neu (fp9x, fp9j)
ⁱ Aliases: txfonts (txr), qfonts/QuasiTimes, T_EX Gyre Termes (qtm)
^j Alias: eulervm (zeur, zeus)

be loaded from the file `microtype.cfg`. You may extend this file with custom settings (or load a different configuration file with the ‘`config`’ option, see section 3.5).

If you are embarking on creating new settings for a font family, you should put them into a separate file, whose name must be: ‘`mt-⟨font family⟩.cfg`’ (e. g., ‘`mt-cmr.cfg`’), and may contain all commands described in the current section 5. These files will be loaded automatically if you are actually using the respective fonts. This package ships with configuration files for a number of font families. Table 3 lists them all.

`\DeclareMicrotypeVariants` {⟨list of suffixes⟩}

`\DeclareMicrotypeVariants*` On its search for a configuration file, the package will also try to remove from the font name a suffix of one or more letters that denotes a ‘variant’ of the base font (cf. Karl Berry’s [Fontname](#)). This allows it to put settings for, e. g., the fonts `padx` (expert set), `padj` (oldstyle numerals) and `pad` (plain) into one and the same file `mt-pad.cfg`. This command expects a comma-separated list of variant suffixes. The starred version appends the suffix(es) to the existing list. The default declaration in `microtype.cfg` is:

```
\DeclareMicrotypeVariants{x,j,w,a,d,0,1}
```

`\DeclareMicrotypeAlias` {} {<alias font>}

This command may be used for fonts that are very similar, or actually the same (for instance if you did not stick to the Berry naming scheme when installing a font). An example would be the Latin Modern fonts, which are derived from Computer Modern, so that it is not necessary to create new settings for them – you could say:

```
\DeclareMicrotypeAlias{lmr}{cmr}
```

which would make the package, whenever it encounters the font `lmr` and does not find settings for it, also try the font `cmr`. In fact, you will find this very line, along with some others, in the default configuration file.

`\LoadMicrotypeFile` {}

In rare cases, it might be necessary to load a font configuration file manually, for instance, from within another configuration file, or to be able to extend settings defined in a file that would otherwise not be loaded automatically, or would be loaded too late.¹⁰ This command will load the file `mt-.cfg`.

6 Context-sensitive setup

The microtype package also allows to apply different micro-typographic settings to the fonts depending on the context in which they occur. This opens up the space for infinite possibilities of tweaking the document's appearance.

`\microtypecontext` {<context assignments>}

This command may be used anywhere in the document (also in the preamble) to change the micro-typographic context in the current group. To each feature (**protrusion**, **expansion**, **tracking**, **spacing** and **kerning**), one context may be assigned. Consequently, only settings with the corresponding 'context' keyword will be applied.

`\begin{microtypecontext}` {<context assignments>}

`\end{microtypecontext}` Like many L^AT_EX commands, it is also available in the form of an environment.

`\textmicrotypecontext` {<context assignments>} {<general text>}

As another possibility, the command `\textmicrotypecontext` sets the context(s) for the text given in the second argument.

Suppose you want the footnote markers in the text to be protruded by a larger amount. You could define settings for the numbers:

```
\SetProtrusion
[ context = footnote ]
{ font      = */*/*/scriptsize } % adapt if necessary
{ 1 = { ,650}, 2 = { ,400}, 3 = { ,400}, 4 = { ,400}, 5 = { ,400},
  6 = { ,400}, 7 = { ,500}, 8 = { ,400}, 9 = { ,400}, 0 = { ,400} }
```

¹⁰ Font package authors might also want to have a look at the hook `\Microtype@Hook`, described in the implementation part, section 14.4.3.

and have the context changed in the footnote marker command. This command differs among the various classes; for the base classes, e.g., `article`, it would be:

```
\newcommand*\new@makefnmark{\hbox{\@textsuperscript{\normalfont
\microtypecontext{protrusion=footnote}\@thefnmark}}}
\renewcommand*\@footnotemark{%
\leavevmode \ifhmode\edef\x@sf{\the\spacefactor}\nobreak\fi
\new@makefnmark \ifhmode\spacefactor\x@sf\fi \relax}
```

For the `memoir` class, you would additionally have to disable auto-detection of multiple footnotes, which prevents protrusion:

```
\renewcommand*\@makefnmark{\hbox{\@textsuperscript{\normalfont
\microtypecontext{protrusion=footnote}\@thefnmark}}}
\let\m@mmf@prepare\relax
\let\m@mmf@check\relax
```

Another possibility would be to employ contexts for a language-dependent setup. For instance, if you are writing a text in French, you could add:

```
\microtypecontext{kerling=french}
```

to the preamble. This would have the effect that kerning settings for the French context would be applied to the document. Should parts of the document be in English, you could write:

```
\textmicrotypecontext{kerling=}{English text!}
```

to reset the context, so that the punctuation characters in these parts will not receive any extra kerning.

Instead of adding these commands manually to your document, you may also load `microtype` with the `babel` option (see section 3.5). The current language will then be automatically detected and the contexts set accordingly.

`\DeclareMicrotypeBabelHook` $\{\langle list\ of\ babel\ languages \rangle\} \{\langle context\ list \rangle\}$

Naturally, `microtype` does not know about the typographic specialties of every language. This command is a means of teaching it how to adjust the context when a particular language is selected. The main configuration file contains among others the following declaration:

```
\DeclareMicrotypeBabelHook
{french,français,acadian,canadien}
{kerling=french, spacing=}
```

Consequently, whenever you switch to the French language, the kerning context will be changed to ‘french’ and the spacing context will be reset. This hook only has an effect if the package was loaded with the `babel` option. Currently, `microtype` supports French and Turkish kerning and English spacing (aka. `\nonfrenchspacing`). For unknown languages, all contexts will be reset.

7 Letterspacing revisited

`\textls` [*amount*] {*general text*}

`\textls*` While the tracking feature, described in section 5.3, will apply to sets of fonts, you may also want to letterspace shorter pieces of text, regardless of the font in which they are typeset.¹¹ For such ad-hoc letterspacing, microtype introduces two commands that can be used in the same way as L^AT_EX's text commands (independently of whether the tracking option is enabled): `\textls` – which also works in math mode – expects the text in the mandatory argument, while `\lsstyle` will switch on letterspacing for all subsequent fonts until the end of the current group. The starred version of `\textls` does not add any extra kerning before or after the text, which may be useful, e. g., for section titles. By default, each character will be spaced out by 100/1000 em = 0.1 em; this amount may be altered in the optional argument to `\textls`, using the `\SetTracking` command, or globally with the `letterspace` package option, with decreasing significance in this order.

`\lslig` {*ligature*}

Since the commands `\textls` and `\lsstyle` will also evaluate the ‘no ligatures’ key for the respective font, you need not worry about protecting or breaking ligatures with most fonts. However, in certain situations, there may be a conflict of ligatures beginning with the same letter, where some of them should be inhibited, while others should not. When letterspacing text typeset in Fraktur fonts, for example, the ligatures ‘ch’, ‘ck’, ‘tz’ and ‘sz’ (‘ß’) should never be broken up; you also usually see the ‘st’ (‘ſt’) ligature in letterspaced text. Furthermore, at least the `yfonts` package realises the short s (‘ſ’) as the ligature ‘s:’. On the other hand, the ‘ct’ ligature and the other ‘long s’ ligatures often found in Fraktur fonts should be suppressed. There are two ways to solve this problem: either don’t disable the ‘s’ and/or ‘c’ ligatures and break those that need to be broken up by inserting ‘`\kern0pt`’ or babel’s “|” shortcut; or disable them and protect those ligatures that need to be protected by enclosing them in the `\lslig` command. So, the following two solutions have the same result (namely, ‘`Ausſichtsloſigkeit`’).

```
\SetTracking[no ligatures={f}]{encoding = LY, family = yfrak}{}
\textfrak{\lsstyle Aus:s{\kern0pt}ichts:los{\kern0pt}igkeit}
```

```
\SetTracking[no ligatures={f,s,c}]{encoding = LY, family = yfrak}{}
\textfrak{\lsstyle Au\lslig{s:}si\lslig{ch}t\lslig{s:}losigkeit}
```

`letterspace.sty` These three commands (plus the `letterspace` option, described in section 3.4) are also available with the alternative `letterspace` package, which is in fact a much stripped-down version of `microtype`, omitting support for all the other extensions (and also omitting the possibilities of the `\SetTracking` command – all ‘f’ ligatures will be disabled, inner and outer spacing and outer kerning will be set to the default values described in section 5.3). If you prefer to forgo `microtype`’s specialties, you

¹¹ Letterspacing should be used cautiously; in particular, letterspacing lower-case text is held in abhorrence by honourable typographers. Unless you know what you are doing, you should probably only letterspace small-capitals or all-capitals. Another just cause may be emphasis in texts typeset in Fraktur fonts.

may load the `letterspace` package instead. Both packages should not be used at the same time.

In contrast to `microtype`, which requires \LaTeX , the `letterspace` package also works with `eplain` or even only `miniltx`: for use with `eplain`, load the package with `\usepackage` inside the `\beginpackages ... \endpackages` environment; with `miniltx` (which does not support package options) simply `\input letterspace.sty`.

8 Disabling ligatures

`\DisableLigatures` [*characters*] [*set of fonts*]

While completely disabling all ligatures of a font (which will also switch off kerning for this font), purposely *lowers* the micro-typographic quality instead of raising it, it is especially useful for typewriter fonts, so that, e. g., in a T1 encoded font, `\texttt{--}` will indeed be printed as `--`, not as `-`. `\DisableLigatures` may be used to specify, in the usual way, a set of fonts for which ligatures should be disabled, for example, of the typewriter font in T1 encoding:

```
\DisableLigatures{encoding = T1, family = tt* }
```

It is also possible to disable selected ligatures only. The optional argument may contain a comma-separated list of characters for which the ligature mechanism should be inhibited:

```
\DisableLigatures[?,!]{encoding = T1} % inhibit ?' and !', but not fi, –, », etc.
```

The character that begins the ligature(s) is what matters. This command may only be used in the preamble, and only once. It requires pdf \TeX 1.30 or newer.

9 Hints and caveats

Use settings that match your font. Although the default settings should give reasonable results for most fonts, the particular font you happen to be using may have different character shapes that necessitate more or less protrusion or expansion. In particular, italic letter shapes may differ wildly in different fonts, hence I have decided against providing default protrusion settings for them.

The file `test-microtype.tex` might be of some help when adjusting the protrusion settings for a font.

Don't use too large a value for expansion. Font expansion is a feature that is supposed to enhance the typographic quality of your document by producing a more uniform greyness of the text block (and potentially reducing the number of necessary hyphenations). When expanding or shrinking a font too much, the effect will be turned into the opposite. Expanding the fonts by more than 2%, i. e., setting a stretch limit of more than 20, should be justified by a typographically trained eye. If you are so lucky as to be in the possession of multiple instances of a Multiple Master font, you may set expansion limits to up to 4%.

Don't use font expansion for web documents (with older pdfTeX versions). With pdfTeX versions older than 1.40, each expanded instance of the font will be embedded in the PDF file, hence the file size may increase by quite large a factor (depending on expansion limits and step). Therefore, courtesy and thriftiness of bandwidth command it not to enable font expansion when creating files to be distributed electronically. With pdfTeX 1.40, which uses a different technique of expansion, the file size increase can be neglected.

Settings for Cyrillic/Greek/Thai etc. encodings are not yet included. The default sets of fonts for which the micro-typographic features will be enabled (see table 2) only contain those encodings for which configurations exist. Therefore, if you are using any other encoding (e. g., T2A, LGR etc.), microtype will not apply to these fonts. You have to define and activate a new font set including the encoding(s) you are using (for details, see section 4). For protrusion at least, you would also have to create settings for the fonts in question (see section 5.1). It goes without saying that contributions for these encodings are more than welcome.

Adjustment of interword spacing is still experimental. The implementation of this feature in pdfTeX is not complete, and may not yield the positive effects on the typographical quality you might expect – in certain situations, there may even be undesired side effects. Therefore, the spacing option should not be chosen blindly; it is also recommended to experiment with the settings in order to understand the workings of this feature.

Only employ kerning adjustment if it is customary in the language's typographic tradition. In contrast to protrusion and expansion, additional kerning does not unconditionally improve the micro-typographical quality of your document. You should only switch it on if you are writing a document in a language whose typographic tradition warrants such kerning. If you are, for example, writing an English text, your readers would probably be rather confused by additional spaces before the punctuation characters.

You might want to disable protrusion in the Table of Contents. In unfortunate situations, enabled protrusion might internally alter the line length in the TOC and similar lists in such a way that an excess leader dot will fit in. The solution is to temporarily disable protrusion for the TOC:

```
\microtypesetup{protrusion=false}  
\tableofcontents  
\microtypesetup{protrusion=true}
```

You might want to disable protrusion in verbatim environments. As you know by now, microtype will by default activate character protrusion for all fonts contained in the font set 'alltext'. This also includes the typewriter font. Although it does make sense to protrude the typewriter font if it appears in running text (like, for example, in this manual), this is probably not desirable inside the verbatim environment. However, microtype has no knowledge about the context that a font appears in but will solely decide by examining its attributes. Therefore, you have to take care of disabling protrusion in verbatim environments for yourself (that

is, if you don't want to disable protrusion for the typewriter font altogether, by choosing a different font set). While the `\microtypesetup` command has of course been designed for cases like this, you might find it tiring to repeat it every time if you are using the `verbatim` environment frequently. The following line, added to the document's preamble, would serve the same purpose:

```
\g@addto@macro\verbatim{\microtypesetup{activate=false}}
```

If you are using the `fancyvrb` or the `listings` package, this is not necessary, since their implementation of the corresponding environments will inhibit protrusion anyway.

Compatibility and interaction with other packages: The `microtype` package is supposed to work happily together with all other L^AT_EX packages (except for `pdfcpot`). However, life isn't perfect, so problems are to be expected. Currently, I am aware of the following issues:

- If you want to use 8-bit characters in the configuration, you have to load the `inputenc` package first. Unicode input is also supported (when loading `inputenc` with the `utf8` or the `utf8x` option). When using multiple input encodings in a document, 8-bit characters in the settings will only work reliably if you specify the `inputenc` key.
- When loading the package with the `babel` option, you must load the `babel` package before `microtype`.
- It is currently not possible to create character-specific settings for Chinese/Japanese/Korean fonts. Therefore, the only micro-typographic extension that can be made to work with the CJK package is font expansion.

Possible error messages and how to get rid of them:

- ! Font csnameendcsname=cmr10+20 at 10.0pt not loadable: Metric (TFM) file not found.
This error message will occur if you are trying to employ font expansion while creating DVI output. Remember, that *automatic* font expansion only works when running pdfT_EX in PDF mode. Although expansion is also possible in DVI mode, it requires that all instances of the expanded fonts exist on your T_EX system.
- ! pdfT_EX error (font expansion): auto expansion is only possible with scalable fonts.
Automatic font expansion has been improved in pdfT_EX 1.40, in that it now not only works with Type 1 fonts but also with TrueType, OpenType and even non-embedded fonts. The above error message indicates either that you are trying to apply expansion to a bitmap (pk) font, which is still not possible, or that the font isn't found at all, e. g., because of missing map entries.

- Warning: pdf_latex: font ptmr8r cannot be expanded (not an included Type1 font)
and the PDF viewer complains about a missing font, e. g., Adobe Reader thusly:
Could not find a font in the Resources dictionary - using Helvetica instead.

With pdfT_EX versions older than 1.40, font expansion can only be applied if the font is actually embedded in the PDF file. If you get the above error message, your T_EX system is not set up to embed (or 'download') the base PostScript fonts (e. g., Times, Helvetica, Courier). In most T_EX distributions, this can be changed in the file `updmap.cfg` by setting `pdftexDownloadBase14` to `true`.

- Warning: pdf_latex (file ecrm1000+20): Font ecrm1000+20 at 1200 not found
Furthermore, pdf_lT_EX versions older than 1.40 require Type 1 fonts for automatic font expansion. When you receive a message like the above, you are probably trying to apply font expansion to a bitmap or TrueType font. With older pdf_lT_EX versions, this is only possible if you manually create expanded instances of the fonts.
- ! Font T1/cmr/m/n/10=ecrm1000 at 10.0pt not loaded: Not enough room left.
Memory parameter ‘font_mem_size’ too small.
- ! TeX capacity exceeded, sorry [maximum internal font number (font_max)=2000].
Memory parameter ‘font_max’ too small.
- ! TeX capacity exceeded, sorry [PDF memory size (pdf_mem_size)=65536].
Memory parameter ‘pdf_mem_size’ too small (pdf_lT_EX versions older than 1.30).
When applying micro-typographic enhancement to a large document with a lot of fonts, pdf_lT_EX may be running out of some kind of memory. It can be increased by setting the respective parameter to a larger value. For web2c-based systems, e. g., T_EX Live, change the settings in texmf.cnf, for MiK_T_EX, in the file miktex.ini (2.4 or older) resp. pdf_latex.ini (2.5 or newer).
- pdf_lTeX warning (font expansion): font should be expanded before its first use
This warning will occur with pdf_lT_EX versions older than 1.40.4, if tracking *and* expansion is applied to a font. It is harmless and can be ignored.

10 Contributions

I would be glad to include configuration files for more fonts. Preparing such configurations is quite a time-consuming task and requires a lot of patience. To alleviate this process, this package also includes a test file that can be used to check at least the protrusion settings (test-microtype.tex).

If you have created a configuration file for another font, or if you have any suggestions for enhancements in the default configuration files, I would gratefully accept them: w.m.l@gmx.net.

11 Acknowledgments

This package would be pointless if *Hàn Thế Thành* hadn’t created the pdf_lT_EX programme in the first place, which introduced the micro-typographic extensions and made them available to the T_EX world. Furthermore, I thank him for helping me to improve this package, and not least for promoting it in [Thành 2004](#) and [Thành 2008](#) and elsewhere. I also thank him and the rest of the pdf_lT_EX team for refuting the idea that T_EX is dead, and for fixing the bugs I find.

Harald Harders has contributed protrusion settings for Adobe Minion. I would also like to thank him for a number of bug reports and suggestions he had to make. *Andreas Bühmann* has suggested the possibility to specify ranges of font sizes, and resourcefully assisted in implementing this. He also came up with some good ideas for the management of complex configurations. *Ulrich Dirr* has

made numerous suggestion, especially concerning the new extensions of interword spacing adjustment and additional character kerning. My thanks also go to *Maciej Eder* for contributing settings for the QX encoding.

I thank *Philipp Lehman* for adding to his *csquotes* package the possibility to restore the original meanings of all activated characters, thus allowing for these characters to be used in the configuration files. *Peter Wilson* kindly provided a hook in his *ledmac/ledpar* packages, so that critical editions can finally also benefit from character protrusion.

Additionally, the following people have reported bugs, made suggestions or helped otherwise (in chronological order): *Tom Kink*, *Herb Schulz*, *Michael Hoppe*, *Gary L. Gray*, *Georg Verweyen*, *Christoph Bier*, *Peter Muthesius*, *Bernard Gaulte* †, *Adam Kucharczyk*, *Mark Rossi*, *Stephan Hennig*, *Michael Zedler*, *Herbert Voß*, *Ralf Stubner*, *Holger Uhr*, *Peter Dyballa*, *Morten Høgholm*, *Steven Bath*, *Daniel Flipo*, *Michalis Miatidis*, *Sven Naumann*, *Ross Hetherington*, *Geoff Vallis*, *Steven E. Harris*, *Karl Berry*, *Peter Meier*, *Nathan Rosenblum*, *Wolfram Schaalo*, *Vasile Gaburici*, *Sveinung Heggen*, *Colin Rourke*, *Maverick Woo* and *Silas S. Brown*.

12 References

Hàn Thế Thành, ‘Micro-typographic extensions to the T_EX typesetting system’, Diss. Masaryk University Brno 2000, in: *TUGBoat*, vol. 21 (2000), no. 4, pp. 317–434. (Online at <http://www.tug.org/TUGboat/Articles/tb21-4/tb69thanh.pdf>)

Hàn Thế Thành, ‘Micro-typographic extensions of pdfT_EX in practice’, in: *TUGBoat*, vol. 25 (2004), no. 1: ‘Proceedings of the Practical T_EX 2004 Conference’, pp. 35–38. (Online at <http://www.tug.org/TUGboat/Articles/tb25-1/thanh.pdf>)

Hàn Thế Thành, ‘Font-specific issues in pdfT_EX’, in: *TUGBoat*, vol. 29 (2008), no. 1: ‘EuroBachOT_EX 2007 Proceedings’, pp. 36–41. (Online at <http://www.tug.org/TUGboat/Articles/tb29-1/tb91thanh-fonts.pdf>)

Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, Hartmut Henkel, Paweł Jackowski, Martin Schröder, *The pdfT_EX user manual*, 25 January 2007. (Available from CTAN at [/systems/pdftex/](http://systems/pdftex/); latest version at <http://sarovar.org/projects/pdftex/>)

Karl Berry, *Fontname. Filenames for T_EX fonts*, September 2005. (Available from CTAN at [/info/fontname/fontname.pdf](http://info/fontname/fontname.pdf))

L^AT_EX3 Project Team, *L^AT_EX2_ε font selection*, 27 November 2005. (Available from CTAN at [/macros/latex/doc/fntguide.pdf](http://macros/latex/doc/fntguide.pdf))

Carsten Schurig, Tobias Schlemmer, *The pdfcp_{rot}.sty package*, 10 June 2005. (Available from CTAN at [/macros/latex/contrib/pdfcp/prot/](http://macros/latex/contrib/pdfcp/prot/))

Melchior Franz, *The soul package*, 17 November 2003. (Available from CTAN at [/macros/latex/contrib/soul/](http://macros/latex/contrib/soul/)). See also Heiko Oberdiek’s extension of this package, *soulutf8*, which adds Unicode support. (Available from CTAN at [/macros/latex/contrib/oberdiek/](http://macros/latex/contrib/oberdiek/))

13 Short history

The comprehensive list of changes can be found in appendix A. The following is a list of all changes relevant in the user land; bug and compatibility fixes are swept under the rug.

2.3d (2009/03/27)

- New default for expansion option ‘step’: 1, if pdfTeX \geq 1.40 [section 3.3]

2.3c (2008/11/11)

- Support for luaTeX enabled by default

2.3 (2007/12/23)

- New key ‘outer kerning’ for \SetTracking to customise outer kerning [section 5.3]
- Adjust protrusion settings for tracking even if protrusion is not enabled
- New option ‘verbose=silent’ to turn all warnings into mere messages [section 3.5]
- The letterspace package also works with eplain or miniltx [section 7]

2.2 (2007/07/14)

- Improvements to tracking/letterspacing: retain kerning (with pdfTeX 1.40.4); automatically adjust protrusion settings
- New key ‘no ligatures’ for \SetTracking to disable selected or all ligatures (with pdfTeX 1.40.4) [section 5.3]
- New keys ‘spacing’ and ‘outer spacing’ for \SetTracking to customise interword spacing [section 5.3]
- Possibility to expand a font with different parameters (with pdfTeX 1.40.4) [section 5.2]
- New optional argument for \DisableLigatures to disable selected ligatures only [section 8]
- New command \DeclareMicrotypeVariants to specify variant suffixes [section 5.7]
- New command \textmicrotypecontext as a wrapper for \microtypecontext [section 6]
- Protrusion settings for Bitstream Letter Gothic

2.1 (2007/01/21)

- New command \slig to protect ligatures in letterspaced text [section 7]

2.0 (2007/01/14)

- Support for the new extensions of pdfTeX version 1.40: tracking/letterspacing, adjustment of interword spacing (glue), and additional kerning (new commands \SetTracking, \SetExtraSpacing, \SetExtraKerning; new options ‘tracking’, ‘spacing’, ‘kerning’) [sections 5.3, 5.4, 5.5]
- New commands \textls and \sstyle for letterspacing, new option ‘letterspace’ [sections 3.4, 7]
- New option ‘babel’ for automatic micro-typographic adjustment to the selected language [sections 3.5, 6]
- New font sets: ‘smallcaps’, ‘footnotesize’, ‘scriptsize’ [section 4; table 2]
- New package ‘letterspace’ providing the commands for robust and hyphenatable letterspacing [section 7]

1.9e (2006/07/28)

- New key ‘inputenc’ to specify the lists’ input encodings [section 5]
- Protrusion settings for Euler math fonts

1.9d (2006/05/05)

- Support for the Central European QX encoding (inheritance, generic protrusion settings, contributed by Maciej Eder; protrusion settings for Times)
- Protrusion settings for various Euro symbol fonts (Adobe, ITC, marvosym)
- Support for Unicode input in the configuration (inputenc/utf8)

1.9c (2006/02/02)

- Protrusion settings for URW Garamond

1.9a (2005/12/05)

- Defer setup until the end of the preamble; consequently, no need to change font defaults before loading microtype, or to put it the other way round, microtype may now be loaded at any time
- Inside the preamble, `\microtypesetup` accepts all package options [section 3.6]
- Protrusion settings for T5 encoded Charter

1.9 (2005/10/28)

- New command `\DisableLigatures` to disable ligatures of fonts (requires pdfTeX version 1.30 or later) [section 8]
- New command `\microtypecontext` to change the configuration context; new key ‘context’ for the configuration commands [section 6]
- New key ‘font’ to add single fonts to the font sets [section 4]
- New key ‘preset’ to set all characters to the specified value before loading the lists
- Value ‘relative’ renamed to ‘character’ for ‘unit’ keys
- Support for the Polish OT4 encoding (protrusion, expansion, inheritance)
- Support for the Vietnamese T5 encoding (protrusion, expansion, inheritance)

1.8 (2005/06/23)

- New command `\DeclareMicrotypeSetDefault` to declare the default font sets [section 4]
- New option ‘config’ to load a different configuration file [section 3.5]
- New option ‘unit’ to measure protrusion factors relative to a dimension instead of the character width [section 5.1]
- Renamed commands from `\..MicroType..` to `\..Microtype..`
- Protrusion settings for AMS math fonts
- Protrusion settings for Times in LY1 encoding completed
- The ‘allmath’ font set also includes U encoding
- When using the `ledmac` package, character protrusion will work for the first time ever (requires pdfTeX version 1.30 or later)

1.7 (2005/03/23)

- Possibility to specify ranges of font sizes in the set declarations and protrusion and expansion settings [sections 4, 5]
- New command `\LoadMicrotypeFile` to load a font configuration file manually [section 5.7]

- Hook `\Microtype@Hook` for font package authors [section 14.4.3]
- New option `'verbose=errors'` to turn all warnings into errors
- Warning when running in draft mode

1.6 (2005/01/24)

- New option `'factor'` to influence protrusion resp. expansion of all characters of a font or font set [sections 3.2, 5]
- When pdf \TeX is too old to expand fonts automatically, expansion has to be enabled explicitly, automatic expansion will be disabled [section 3.1]
- Use e- \TeX extensions, if available

1.5 (2004/12/15)

- When output mode is DVI, font expansion has to be enabled explicitly, automatic expansion will be disabled [section 3.1]
- New option `'selected'` to enable selected expansion, default: false [sections 3.3, 5.2]
- New default for expansion option `'step'`: 4 ($\min(\text{stretch}, \text{shrink})/5$) [section 3.3]
- Protrusion settings for Bitstream Charter

1.4 (2004/11/12)

- Set up fonts independently from \LaTeX font loading
- New option: `'final'` [section 3.5]

1.2 (2004/10/03)

- New font sets: `'allmath'` and `'basicmath'` [section 4; table 2]
- Protrusion settings for Computer Modern Roman math symbols
- Protrusion settings for TS1 encoding completed for Computer Modern Roman and Adobe Garamond

1.1 (2004/09/21)

- Protrusion settings for Adobe Minion, contributed by Harald Harders
- New command: `\DeclareCharacterInheritance` [section 5.6]
- Characters may also be specified as octal or hexadecimal numbers [section 5]

1.0 (2004/09/11)

- First CTAN release

14 Implementation

The docstrip modules in this file are:

`driver`: The documentation driver, only visible in the `dtx` file.

`package`: The code for the microtype package (`microtype.sty`).

`letterspace`: The code for the `letterspace` package (`letterspace.sty`).

`lua`: Code for `luaTeX` (microtype only).

`plain`: Code for `eplain`, `miniltx` (`letterspace` only).

`debug`: Code for additional output in the log file.

Used for – surprise! – debugging purposes.

`config`: Surrounds all configuration modules.

`cfg-t`: Surrounds (Latin) text configurations.

`m-t`: The main configuration file (`microtype.cfg`).

`bch`: Settings for Bitstream Charter (`mt-bch.cfg`).

`blg`: Settings for Bitstream Letter Gothic (`mt-blg.cfg`).

`cmr`: Settings for Computer Modern Roman (`mt-cmr.cfg`).

`pad`: Settings for Adobe Garamond (`mt-pad.cfg`).

`ppl`: Settings for Palatino (`mt-ppl.cfg`).

`ptm`: Settings for Times (`mt-ptm.cfg`).

`pmn`: Settings for Adobe Minion (`mt-pmn.cfg`).

Contributed by *Harald Harders*.

`ugm`: Settings for URW Garamond (`mt-ugm.cfg`).

`cfg-u`: Surrounds non-text configurations (U encoding).

`msa`: Settings for AMS ‘a’ symbol font (`mt-msa.cfg`).

`msb`: Settings for AMS ‘b’ symbol font (`mt-msb.cfg`).

`euf`: Settings for Euler Fraktur font (`mt-euf.cfg`).

`eur`: Settings for Euler Roman font (`mt-eur.cfg`).

`eus`: Settings for Euler Script font (`mt-eus.cfg`).

`cfg-e`: Surrounds Euro symbol configurations.

`zpeu`: Settings for Adobe Euro symbol fonts (`mt-zpeu.cfg`).

`euroitc`: Settings for ITC Euro symbol fonts (`mt-euroitc.cfg`).

`mvs`: Settings for marvosym Euro symbol (`mt-mvs.cfg`).

`test`: A helper file that may be used to create and test protrusion settings (`test-microtype.tex`).

And now for something completely different.

¹ `<package|letterspace>`

14.1 Preliminaries

`\MT@MT` This is us.

```
2 \def\MT@MT
3 <package> {microtype}
4 <letterspace> {letterspace}
```

`\MT@fix@catcode` We have to make sure that the category codes of some characters are correct (the german package, for instance, makes " active). Probably overly cautious. Ceterum censeo: it should be forbidden for packages to change catcodes within the preamble.

`\MT@restore@catcodes` Polite as we are, we'll restore them afterwards.

```
5 \let\MT@restore@catcodes\@empty
6 \def\MT@fix@catcode#1#2{%
7   \edef\MT@restore@catcodes{%
8     \MT@restore@catcodes
9     \catcode#1 \the\catcode#1\relax
10  }%
11  \catcode#1 #2\relax
12 }
13 <package>\MT@fix@catcode{17}{14}% ^^Q (comment)
14 \MT@fix@catcode{24}{9}% ^^X (ignore)
15 <package>\MT@fix@catcode{33}{12}% !
16 <package>\MT@fix@catcode{34}{12}% "
17 \MT@fix@catcode{36}{3}% $ (math shift)
18 \MT@fix@catcode{39}{12}% '
19 \MT@fix@catcode{42}{12}% *
20 \MT@fix@catcode{43}{12}% +
21 \MT@fix@catcode{44}{12}% ,
22 \MT@fix@catcode{45}{12}% -
23 \MT@fix@catcode{58}{12}% :
24 \MT@fix@catcode{60}{12}% <
25 \MT@fix@catcode{61}{12}% =
26 \MT@fix@catcode{62}{12}% >
27 <package>\MT@fix@catcode{63}{12}% ?
28 \MT@fix@catcode{94}{7}% ^ (superscript)
29 \MT@fix@catcode{96}{12}% ~
30 <package>\MT@fix@catcode{124}{12}% |
```

These are all commands for the outside world. We define them here as blank commands, so that they won't generate an error if we are not running pdf \TeX .

```
31 <*package>
32 \newcommand*\DeclareMicrotypeSet[3]{}{}
33 \newcommand*\UseMicrotypeSet[2]{}{}
34 \newcommand*\DeclareMicrotypeSetDefault[2]{}{}
35 \newcommand*\SetProtrusion[3]{}{}
36 \newcommand*\SetExpansion[3]{}{}
37 \newcommand*\SetTracking[3]{}{}
38 \newcommand*\SetExtraKerning[3]{}{}
39 \newcommand*\SetExtraSpacing[3]{}{}
40 \newcommand*\DisableLigatures[2]{}{}
41 \newcommand*\DeclareCharacterInheritance[3]{}{}
42 \newcommand*\DeclareMicrotypeVariants[1]{}
43 \newcommand*\DeclareMicrotypeAlias[2]{}
44 \newcommand*\LoadMicrotypeFile[1]{}
45 \newcommand*\DeclareMicrotypeBabelHook[2]{}
46 \newcommand*\microtypesetup[1]{}
47 \newcommand*\microtypecontext[1]{}
48 \newcommand*\textmicrotypecontext[2]{#2}
49 \@ifpackageloaded{letterspace}{\let\MT@textls\relax}{%
50 </package>
```

```

51 \newcommand*\lststyle{}
52 \newcommand\textls[2][]{\textls[2]{}}
53 \def\textls#1#{}
54 \newcommand*\lslig[1]{#1}
55 *package
56 }

```

These commands also have a starred version.

```

57 \def\DeclareMicrotypeSet#1#{\@gobbletwo}
58 \def\DeclareMicrotypeVariants#1#{\@gobble}

```

Set declarations are only allowed in the preamble (resp. the main configuration file). The configuration commands, on the other hand, must be allowed in the document, too, since they may be called inside font configuration files, which, in principle, may be loaded at any time.

```

59 \@onlypreamble\DeclareMicrotypeSet
60 \@onlypreamble\UseMicrotypeSet
61 \@onlypreamble\DeclareMicrotypeSetDefault
62 \@onlypreamble\DisableLigatures
63 \@onlypreamble\DeclareMicrotypeVariants
64 \@onlypreamble\DeclareMicrotypeBabelHook

```

`\MT@old@cmd` The old command names had one more hunch.

```

65 \def\MT@old@cmd#1#2{%
66   \newcommand*#1{\MT@warning{%
67     \string#1 is deprecated. Please use\MessageBreak
68     \string#2 instead}%
69   \let #1#2#2}}
70 \MT@old@cmd\DeclareMicroTypeAlias\DeclareMicrotypeAlias
71 \MT@old@cmd\DeclareMicroTypeSet \DeclareMicrotypeSet
72 \MT@old@cmd\UseMicroTypeSet \UseMicrotypeSet
73 \MT@old@cmd\LoadMicroTypeFile \LoadMicrotypeFile
74 /package

```

`\MT@warning` Communicate.

```

\MT@warning@nl 75 \def\MT@warning{\PackageWarning\MT@MT}
\MT@info       76 \def\MT@warning@nl#1{\MT@warning{#1\@gobble}}
\MT@info@nl    77 *package
\MT@vinfo      78 \def\MT@info{\PackageInfo\MT@MT}
\MT@vinfo      79 \def\MT@info@nl#1{\MT@info{#1\@gobble}}
\MT@error      80 \let\MT@vinfo@gobble
\MT@warn@err   81 \def\MT@error{\PackageError\MT@MT}
\MT@warn@err   82 \def\MT@warn@err#1{\MT@error{#1}{%
83   This error message appears because you loaded the \MT@MT'\MessageBreak
84   package with the option `verbose=errors'. Consult the documentation\MessageBreak
85   in \MT@MT.pdf to find out what went wrong.}}

```

14.1.1 Debugging

`\tracingmicrotype` Cases for `\tracingmicrotype`:

- `\MT@dinfo` 0: almost none
- `\MT@dinfo@nl` 1: + sets & lists
- 2: + heirs
- 3: + slots
- 4: + factors

```

86 (*debug)
87 \MT@warning@n1{This is the debug version}
88 \newcount\tracingmicrotype
89 \tracingmicrotype=2
90 \def\MT@info#1{\PackageInfo\MT@MT{#1}\MT@addto@annot{#1}}
91 \def\MT@info@n1#1{\PackageInfo\MT@MT{#1}\@gobble}\MT@addto@annot{#1}}
92 \let\MT@vinfo\MT@info@n1
93 \def\MT@warning#1{\PackageWarning\MT@MT{#1}\MT@addto@annot{Warning: #1}}
94 \def\MT@warning@n1#1{\PackageWarning\MT@MT{#1}\@gobble}\MT@addto@annot{Warning: #1}}
95 \def\MT@dinfo#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info{#2}\fi}
96 \def\MT@dinfo@n1#1#2{\ifnum\tracingmicrotype<#1 \else\MT@info@n1{#2}\fi}

```

\tracingmicrotypeinpdf

Another debug method: font switches can be marked in the PDF file with a small caret, an accompanying popup text box displaying all debug messages.

Cases for \tracingmicrotypeinpdf:

- 1: show new fonts
- 2: + show known fonts

```
97 \newcount\tracingmicrotypeinpdf
```

Let's see how it works ...

```
\tracingmicrotypeinpdf=2
```

```

\MT@pdf@annot
\MT@addto@annot
\ifMT@inannot

```

During font setup, we save the text for the popup in \MT@pdf@annot. (This requires pdf_{TeX} ≥ 1.30.) The pdftexcmds package provides pdf_{TeX}'s utility commands in lua_{TeX}, too.

```

98 \RequirePackage{pdftexcmds}
99 \newif\ifMT@inannot \MT@inannottrue
100 \let\MT@pdf@annot\empty
101 \def\MT@addto@annot#1{\ifnum\tracingmicrotypeinpdf>\z@ \ifMT@inannot
102   {\def\MessageBreak{^^J@spaces}%
103    \MT@xadd\MT@pdf@annot{\pdf@escapestring{#1^^J}}}\fi\fi}

```

\iftracingmicrotypeinpdfall

With \tracingmicrotypeinpdfall false, the PDF output is (hopefully) identical, but some font switches will not be displayed; otherwise the output is affected, but *all* font switches are visible. In the latter case, we also insert a small kern so that multiple font switches are discernable.

```
104 \newif\iftracingmicrotypeinpdfall
```

\MT@show@pdfannot

A red caret is shown for fonts which are actually set up by *Microtype*, a green one marks fonts that we have already seen. The /Caret annotation requires a viewer for PDF version 1.5 (you could use /Text if you're using an older PDF viewer).

```

105 \def\MT@show@pdfannot#1{%
106   \ifnum\tracingmicrotypeinpdf<#1 \else
107     \iftracingmicrotypeinpdfall\leavevmode\fi
108     \pdfannot height 4pt width 4pt depth 2pt {%
109       /Subtype/Caret
110       /T(\expandafter\string\font@name)
111       \ifcase#1\or
112         /Subj(New font)/C[1 0 0]
113       \else
114         /Subj(Known font)/C[0 1 0]
115       \fi
116       /Contents(\MT@pdf@annot)
117     }%
118     \iftracingmicrotypeinpdfall\kern1pt \fi
119     \global\MT@inannotfalse
120   \fi

```

```

121 }
122 </debug>
123 </package>

```

14.1.2 Requirements

`\MT@plain` The letterspace package works with:

0: `miniltx`

1: `eplain`

2: `LATEX`

For plain usage, we have to copy some commands from `latex.ltx`.

```

124 <*plain>
125 \def\MT@plain{2}
126 \ifx\documentclass\@undefined
127   \def\MT@plain{1}
128   \def\hmode\bgroup{\leavevmode\bgroup}
129   \def\nfss@text#1{{\mbox{#1}}}
130   \let\@typeset@protect\relax
131   \ifx\eplain\@undefined
132     \def\MT@plain{0}
133     \def\PackageWarning#1#2{%
134       \begingroup
135         \newlinechar=10 %
136         \def\MessageBreak{^^J(#1)\@spaces\@spaces\@spaces\@spaces}%
137         \immediate\write16{^^JPackage #1 Warning: #2\on@line.^^J}%
138       \endgroup
139     }
140     \def\on@line{ on input line \the\inputlineno}
141     \def\@spaces{\space\space\space\space}
142   \fi
143 \fi

```

`\MT@requires@latex` Better use groups than plain ifs.

```

144 \def\MT@requires@latex#1{%
145   \ifnum\MT@plain<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
146 }
147 </plain>

```

`\MT@pdftex@no` pdf_TE_X's features for which we provide an interface here haven't always been available, and some specifics have changed over time. Therefore, we have to test which pdf_TE_X we're using, if any. `\MT@pdftex@no` will be used throughout the package to respectively do the right thing.

Currently, we have to distinguish seven cases for pdf_TE_X:

0: not running pdf_TE_X

1: pdf_TE_X (< 0.14f)

2: + micro-typographic extensions (0.14f,g)

3: + protrusion relative to 1 em (\geq 0.14h)

4: + automatic font expansion; protrusion no longer has to be set up first; scale factor fixed to 1000; default `\efcode` = 1000 (\geq 1.20)

5: + `\(left,right)marginkern`; `\pdfnoligatures`; `\pdfstrcmp`; `\pdfescapestring` (\geq 1.30)

6: + adjustment of interword spacing; extra kerning; `\letterspacefont`; `\pdfmatch`¹²; `\pdftracingfonts`; always e-TeX (≥ 1.40)

7: + `\letterspacefont` doesn't disable ligatures and kerns; `\pdfcopyfont` ($\geq 1.40.4$)

```
148 \def\MT@pdftex@no{0}
```

A hack circumventing the TeX Live 2004 hack which undefines the pdfTeX primitives in the format in order to hide the fact that pdfTeX is being run from the user. This has been *fixed* in TeX Live 2005.

```
149 \ifx\normalpdftexversion\@undefined \else
150   \let\pdftexversion\normalpdftexversion
151   \let\pdftexrevision\normalpdftexrevision
152   \let\pdfoutput\normalpdfoutput
153 \fi
```

Old packages might have let `\pdftexversion` to `\relax`.

```
154 \ifx\pdftexversion\@undefined \else
155   \ifx\pdftexversion\relax \else
156     <debug>\MT@info@nl{0}{this is pdftex \the\pdftexversion(\pdftexrevision)}
157     \def\MT@pdftex@no{7}
158     <*package>
159     \ifnum\pdftexversion = 140
160       \ifnum\pdftexrevision < 4
161         \def\MT@pdftex@no{6}
162       \fi
163     \else
164     </package>
165       \ifnum\pdftexversion < 140
166         \def\MT@pdftex@no{5}
167     <*package>
168         \ifnum\pdftexversion < 130
169           \def\MT@pdftex@no{4}
170         \ifnum\pdftexversion < 120
171           \def\MT@pdftex@no{3}
172         \ifnum\pdftexversion = 14
173           \ifnum\expandafter`\pdftexrevision < `h
174             \def\MT@pdftex@no{2}
175           \ifnum\expandafter`\pdftexrevision < `f
176             \def\MT@pdftex@no{1}
177           \fi
178         \fi
179       \else
180         \ifnum\pdftexversion < 14
181           \def\MT@pdftex@no{1}
182         \fi
183       \fi
184     \fi
185   \fi
186 \fi
187 </package>
188 \fi
189 \fi
190 \fi
191 <debug>\MT@info@nl{0}{pdftex no.: \MT@pdftex@no}
```

`\MT@clear@options` If we are not using pdfTeX or in case it is too old, we disable everything and exit.

```
192 \def\MT@clear@options{%
193   <plain> \MT@requires@latex1{%
194     \AtEndOfPackage{\let\unprocessedoptions\relax}%
```

12 This command was actually introduced in 1.30, but failed on strings longer than 1023 bytes.

```

195 \let\CurrentOption\@empty
196 <plain> }\relax
197 }

198 \ifnum\MT@pdftex@no <
199 <package> 2
200 <letterspace> 6
201 \MT@warning@n1{You
202 \ifcase\MT@pdftex@no
203 don't seem to be using pdftex.\MessageBreak
204 ~\MT@MT' only works with pdftex.\MessageBreak
205 Try running ~pdflatex' instead of
206 ~\ifx\XeTeXversion\@undefined\else xe\fi latex'%
207 \else
208 are using a pdftex version older than
209 <package> 0.14f%
210 <letterspace> 1.40%
211 .\MessageBreak
212 ~\MT@MT' does not work with this version.\MessageBreak
213 Please install a newer version of pdftex%
214 \fi
215 }
216 \MT@clear@options\MT@restore@catcodes
217 \endinput\fi

```

Since luaTeX is included in TeX Live 2008, we now support it by default, even though it's still experimental. Letterspacing doesn't work at all yet, since luaTeX doesn't know the \letterspacefont command.

```

218 <lua>|letterspace>
219 \ifx\directlua\@undefined \else
220 \ifx\directlua\relax \else
221 <letterspace> \MT@error
222 <letterspace> \MT@warning@n1
223 {~\MT@MT'
224 <letterspace> only works with luatex if you generate%
225 <letterspace> doesn't currently work with luatex.%
226 \MessageBreak
227 <letterspace> the package with the ~lua' option%
228 <letterspace> Bye bye%
229 }
230 <letterspace> {}
231 <letterspace> \MT@clear@options\MT@restore@catcodes
232 <letterspace> \expandafter\expandafter\expandafter\endinput
233 \fi
234 \fi
235 </lua>|letterspace>

```

Still there? Then we can begin: We need the keyval package, including the 'new' \KV@@sp@def implementation.

```

236 \RequirePackage{keyval}[1997/11/10]
237 <*package>

```

\MT@toks We need a token register.

```

238 \newtoks\MT@toks

```

\ifMT@if@ A scratch if.

```

239 \newif\ifMT@if@

```

14.1.3 Declarations

```

\ifMT@protrusion      These are the global switches ...
\ifMT@expansion 240 \newif\ifMT@protrusion
\ifMT@auto 241 \newif\ifMT@expansion
\ifMT@selected 242 \newif\ifMT@auto
\ifMT@selected 243 \newif\ifMT@selected
\ifMT@noligatures 244 \newif\ifMT@noligatures
\ifMT@draft 245 \newif\ifMT@draft
\ifMT@spacing 246 \newif\ifMT@spacing
\ifMT@kerning 247 \newif\ifMT@kerning
\ifMT@tracking 248 \newif\ifMT@tracking
\ifMT@tracking 249 \newif\ifMT@babel
\MT@pr@babel ... and numbers.
\MT@ex@level 250 \let\MT@pr@level\tw@
\MT@pr@factor 251 \let\MT@ex@level\tw@
\MT@ex@factor 252 \let\MT@pr@factor\m@ne
\MT@ex@factor 253 \let\MT@ex@factor\m@ne
\MT@sp@factor 254 \let\MT@sp@factor\m@ne
\MT@kn@factor 255 \let\MT@kn@factor\m@ne

\MT@pr@unit      Default unit for protrusion settings is character width, for spacing space, for kerning
\MT@sp@unit      (and tracking) 1 em.
\MT@kn@unit 256 \let\MT@pr@unit\@empty
                257 \let\MT@sp@unit\m@ne
                258 \def\MT@kn@unit{1em}

\MT@stretch      Expansion settings.
\MT@shrink 259 \let\MT@stretch\m@ne
\MT@step 260 \let\MT@shrink \m@ne
          261 \let\MT@step \m@ne

\MT@pr@min      Minimum and maximum values allowed by pdfTeX.
\MT@pr@max 262 \def\MT@pr@min{-\@m}
\MT@ex@min 263 \let\MT@pr@max\@m
\MT@ex@min 264 \let\MT@ex@min\z@
\MT@ex@max 265 \let\MT@ex@max\@m
\MT@sp@min 266 \def\MT@sp@min{-\@m}
\MT@sp@max 267 \let\MT@sp@max\@m
\MT@kn@min 268 \def\MT@kn@min{-\@m}
\MT@kn@min 269 \let\MT@kn@max\@m
\MT@kn@max 270 /package
\MT@tr@min 271 \def\MT@tr@min{-\@m}
\MT@tr@min 272 \let\MT@tr@max\@m
\MT@tr@max 273 *package

\MT@factor@default      Default factor.
                        274 \def\MT@factor@default{1000 }

\MT@stretch@default      Default values for expansion.
\MT@shrink@default 275 \def\MT@stretch@default{20 }
\MT@step@default 276 \def\MT@shrink@default{20 }
                  277 \def\MT@step@default{4 }

\MT@letterspace      Default value for letterspacing (in thousandths of 1 em).
\MT@letterspace@default 278 /package
                        279 \let\MT@letterspace\m@ne
                        280 \def\MT@letterspace@default{100}
                        281 *package

```


`\ifMT@document` Our private test whether we're still in the preamble.

282 `\newif\ifMT@document`

14.1.4 Auxiliary macros

`\MT@maybe@etex` For definitions that depend on e-TeX features.

```
283 \ifcase 0%
284   \ifx\TeXversion\undefined 1\else
285   \ifx\TeXversion\relax 1\else
286   \ifcase\TeXversion 1\fi
287   \fi
288 \fi
289 \else
290   \catcode\^^Q=9 \catcode\^^X=14
291 \fi
292 <debug>\MT@info{n1}{0}{this is
293 <debug>\^^Q not
294 <debug> etex}
```

`\MT@requires@pdftex` For definitions that depend on a particular pdfTeX version.

```
295 \def\MT@requires@pdftex#1{%
296   \ifnum\MT@pdftex@no<#1 \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
297 }
298 <debug>\MT@requires@pdftex6{\pdftracingfonts=1}\relax
```

`\MT@requires@luatex` For definitions that depend on luaTeX.

```
299 <*lua>
300 \let\MT@requires@luatex\@secondoftwo
301 \ifx\directlua\undefined \else
302   \ifx\directlua\relax \else
303     \let\MT@requires@luatex\@firstoftwo
304   \fi
305 \fi
306 <debug>\MT@info{n10}{this is \MT@requires@luatex}{\not }luatex}
```

`\MT@lua` Communicate with lua.

Beginning with luaTeX 0.36, `\directlua` no longer requires a state number.

```
307 \MT@requires@luatex{
308   \ifnum\luatexversion<36
309     \def\MT@lua{\directlua0}
310   \else
311     \def\MT@lua{\directlua}
312   \fi
313 }\relax
314 </lua>
315 </package>
```

`\MT@glet` The forgotten primitive.

316 `\def\MT@glet{\global\let}`

`\MT@exp@cs` Commands to create command sequences. Those that are going to be defined globally should be created inside a group so that the save stack won't explode.

`\MT@exp@gcs`

```
317 \def\MT@exp@cs#1#2{\expandafter#1\csname#2\endcsname}
318 <*package>
319 \def\MT@exp@gcs#1#2{\begingroup\expandafter\endgroup\expandafter#1\csname#2\endcsname}
```

`\MT@def@n` This is `\@namedef` and global.

```
\MT@gdef@n 320 \def\MT@def@n{\MT@exp@cs\def}
321 \def\MT@gdef@n{\MT@exp@gcs\gdef}
```

```

\MT@edef@n    Its expanding versions.
\MT@xdef@n 322 </package>
           323 \def\MT@edef@n{\MT@exp@cs\edef}
           324 <*package>
           325 \def\MT@xdef@n{\MT@exp@gcs\xdef}

\MT@let@nc    \let a \csname sequence to a command.
\MT@glet@nc 326 \def\MT@let@nc{\MT@exp@cs\let}
           327 \def\MT@glet@nc{\MT@exp@gcs\MT@glet}

\MT@let@cn    \let a command to a \csname sequence.
           328 \def\MT@let@cn#1#2{\expandafter\let\expandafter#1\csname #2\endcsname}

\MT@let@nn    \let a \csname sequence to a \csname sequence.
\MT@glet@nn 329 \def\MT@let@nn{\MT@exp@cs\MT@let@cn}
           330 \def\MT@glet@nn{\MT@exp@gcs{\global\expandafter\MT@let@cn}}

\MT@@font    Remove trailing space from the font name.
           331 \def\MT@@font{\expandafter\string\MT@font}

\MT@exp@one@n Expand the second token once and enclose it in braces.
           332 </package>
           333 \def\MT@exp@one@n#1#2{\expandafter#1\expandafter{#2}}

\MT@exp@two@c Expand the next two tokens after <#1> once.
           334 \def\MT@exp@two@c#1{\expandafter\expandafter\expandafter#1\expandafter}
           335 <*package>

\MT@exp@two@n Expand the next two tokens after <#1> once and enclose them in braces.
           336 \def\MT@exp@two@n#1#2#3{%
           337   \expandafter\expandafter\expandafter
           338   #1\expandafter\expandafter\expandafter
           339   {\expandafter#2\expandafter}\expandafter{#3}}

You do not wonder why \MT@exp@one@c doesn't exist, do you?

\MT@ifdefined@c@T Wrapper for testing whether command resp. \csname sequence is defined. If we
\MT@ifdefined@c@TF are running e-TeX, we will use its primitives \ifdefined and \ifcsname, which
\MT@ifdefined@n@T decreases memory use substantially.
\MT@ifdefined@n@TF 340 \def\MT@ifdefined@c@T#1{%
           341   ^^X \ifdefined#1\expandafter\@firstofone\else\expandafter\@gobble\fi
           342   ^^Q \ifx#1\@undefined\expandafter\@gobble\else\expandafter\@firstofone\fi
           343 }
           344 </package>
           345 \def\MT@ifdefined@c@TF#1{%
           346   ^^X \ifdefined#1\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
           347   <package>^^Q \ifx#1\@undefined
           348   <package>^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
           349 }
           350 \def\MT@ifdefined@n@T#1{%
           351   ^^X \ifcsname#1\endcsname\expandafter\@firstofone\else\expandafter\@gobble\fi
           352   <package>^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
           353   <package>^^Q \expandafter\@gobble\else\expandafter\@firstofone\fi
           354 }
           355 <*package>
           356 \def\MT@ifdefined@n@TF#1{%
           357   ^^X \ifcsname#1\endcsname\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
           358   ^^Q \begingroup\MT@exp@two@c\endgroup\ifx\csname #1\endcsname\relax
           359   ^^Q \expandafter\@secondoftwo\else\expandafter\@firstoftwo\fi
           360 }

```

`\MT@detokenize@n` Translate a macro into a token list. With e-TeX, we can use `\detokenize`. We also need to remove the last trailing space; and only the last one – therefore the fiddling (and the `\string` isn't perfect, of course).

```

361 \def\MT@detokenize@n#1{%
362   ^^X \expandafter\MT@rem@last@space\detokenize{#1} \@nil
363   ^^Q \string#1%
364 }
365 \def\MT@detokenize@c#1{%
366   ^^X \MT@exp@one@n\MT@detokenize@n#1%
367   ^^Q \MT@exp@two@c\MT@rem@last@space\strip@prefix\meaning#1 \@nil
368 }
369 \def\MT@rem@last@space#1 #2{#1%
370   \ifx\@nil#2\else \space
371   \expandafter\MT@rem@last@space\expandafter#2\fi
372 }
```

`\MT@ifempty` Test whether argument is empty.

```

373 </package>
374 \begingroup
375 \catcode`\%=12
376 \catcode`\&=14
377 \gdef\MT@ifempty#1{%
378   \if %#1%&
379     \expandafter\@firstoftwo
380   \else
381     \expandafter\@secondoftwo
382   \fi
383 }
384 \endgroup
385 <*package>
```

`\MT@ifint` Test whether argument is an integer, using an old trick by Mr. Arseneau, or the latest and greatest from pdfTeX or luaTeX (which also allows negative numbers, as required by the `letterspace` option).

```

386 \MT@requires@pdftex6{
387   <*lua>
388   \MT@requires@luatex{
389     \def\MT@ifint#1{%
390       \csname \MT@lua{
391         if string.find("\luaescapestring{#1}", "^-[0-9]+ *$")
392         then tex.write("@firstoftwo")
393         else tex.write("@secondoftwo")
394       end}%
395     \endcsname
396   }
397 }{
398 </lua>
399 </package>
400   \def\MT@ifint#1{%
401     \ifcase\pdfmatch{^-[0-9]+ *$}{#1}\relax
402     \expandafter\@secondoftwo
403   \else
404     \expandafter\@firstoftwo
405   \fi
406 }
407 <*package>
408 <lua> }
409 }{
410   \def\MT@ifint#1{%
411     \if!\ifnum9<1#1!\else?\fi
412     \expandafter\@firstoftwo
```

```

413     \else
414     \expandafter\@secondoftwo
415     \fi
416   }
417 }

\MT@ifdimen    Test whether argument is dimension (or number). (nd and nc are new Didot resp.
                Cicero, added in pdfTeX 1.30; px is a pixel.)
418 \MT@requires@pdftex6{
419   \lua
420   \MT@requires@luatex{
421     \def\MT@ifdimen#1{%
422       \csname \MT@lua{
423         if (string.find("\luaescapestring{#1}", "^-[0-9]+(\@percentchar a*) *$") or
424           string.find("\luaescapestring{#1}", "^-[0-9]*[.][0-9]+(\@percentchar a*) *$"))
425         then tex.write("@firstoftwo")
426         else tex.write("@secondoftwo")
427       end}%
428     \endcsname
429   }
430   }{
431     \lua
432     \def\MT@ifdimen#1{%
433       \ifcase\pdfmatch{^[0-9]+([.][0-9]+)?|.[.][0-9]+)%
434         (em|ex|cm|mm|in|pc|pt|dd|cc|bp|sp|nd|nc|px)? *$}{#1}\relax
435       \expandafter\@secondoftwo
436     \else
437       \expandafter\@firstoftwo
438     \fi
439   }
440   \lua }
441 }{
442   \def\MT@ifdimen#1{%
443     \setbox\z@=\hbox{%
444       \MT@count=1#1\relax
445       \ifnum\MT@count=\@ne
446         \aftergroup\@secondoftwo
447       \else
448         \aftergroup\@firstoftwo
449       \fi
450     }%
451   }
452 }

\MT@ifdim      Test floating point numbers.
453 \def\MT@ifdim#1#2#3{%
454   \ifdim #1\p@ #2 #3\p@
455     \expandafter\@firstoftwo
456   \else
457     \expandafter\@secondoftwo
458   \fi
459 }

\MT@ifstreq    Test whether two strings (fully expanded) are equal.
460 \MT@requires@pdftex5{
461   \lua
462   \MT@requires@luatex{
463     \def\MT@ifstreq#1#2{%
464       \csname \MT@lua{
465         if "\luaescapestring{#1}" == "\luaescapestring{#2}"
466         then tex.write("@firstoftwo")
467         else tex.write("@secondoftwo")

```

```

468     end}%
469     \endcsname
470   }
471 }{
472 /lua
473   \def\MT@ifstreq#1#2{%
474     \ifcase\pdfstrcmp{#1}{#2}\relax
475     \expandafter\@firstoftwo
476     \else
477     \expandafter\@secondoftwo
478     \fi
479   }
480 lua }
481 }{
482   \def\MT@ifstreq#1#2{%
483     \edef\MT@res@a{#1}%
484     \edef\MT@res@b{#2}%
485     \ifx\MT@res@a\MT@res@b
486     \expandafter\@firstoftwo
487     \else
488     \expandafter\@secondoftwo
489     \fi
490   }
491 }

\MT@xadd    Add item to a list.
492 \def\MT@xadd#1#2{%
493   \ifx#1\relax
494   \xdef#1{#2}%
495   \else
496   \xdef#1{#1#2}%
497   \fi
498 }

\MT@xaddb   Add item to the beginning.
499 \def\MT@xaddb#1#2{%
500   \ifx#1\relax
501   \xdef#1{#2}%
502   \else
503   \xdef#1{#2#1}%
504   \fi
505 }

\MT@map@clist@n    Run <#2> on all elements of the comma list <#1>. This and the following is modelled
\MT@map@clist@c    after LATEX3 commands.
\MT@map@clist@     506 /package
\MT@clist@function 507 \def\MT@map@clist@n#1#2{%
508   \ifx\@empty#1\else
\MT@clist@break    509   \def\MT@clist@function##1{#2}%
510     \MT@map@clist@#1,\@nil,\@nnil
511     \fi
512   }
513   \def\MT@map@clist@c#1{\MT@exp@one@n\MT@map@clist@n#1}
514   \def\MT@map@clist@#1,{%
515     \ifx\@nil#1%
516     \expandafter\MT@clist@break
517     \fi
518     \MT@clist@function{#1}%
519     \MT@map@clist@
520   }
521   \let\MT@clist@function\@gobble
522   \def\MT@clist@break#1\@nnil{}
```

```

523 (*package)

\MT@map@tlist@n      Execute <#2> on all elements of the token list <#1>. \MT@tlist@break can be used
\MT@map@tlist@c      to jump out of the loop.
\MT@map@tlist@       524 \def\MT@map@tlist@n#1#2{\MT@map@tlist@#2#1\@nnil}
\MT@tlist@break      525 \def\MT@map@tlist@c#1#2{\expandafter\MT@map@tlist@\expandafter#2#1\@nnil}
                    526 \def\MT@map@tlist@#1#2{%
                    527   \ifx\@nnil#2\else
                    528     #1{#2}%
                    529     \expandafter\MT@map@tlist@
                    530     \expandafter#1%
                    531   \fi
                    532 }
                    533 \def\MT@tlist@break#1\@nnil{\fi}

\ifMT@inlist@        Test whether item <#1> is in comma list <#2>. Using \pdfmatch would be slower.
\MT@in@clist         534 \newif\ifMT@inlist@
                    535 \def\MT@in@clist#1#2{%
                    536   \def\MT@res@a#1,#1,##2##3\@nnil{%
                    537     \ifx##2\@empty
                    538       \MT@inlist@false
                    539     \else
                    540       \MT@inlist@true
                    541     \fi
                    542   }%
                    543   \expandafter\MT@res@a\expandafter,#2,#1,\@empty\@nnil
                    544 }

\MT@rem@from@clist   Remove item <#1> from comma list <#2>. This is basically \@removeelement from
                    ltcntrl.dtx. Using \pdfmatch and \pdflastmatch here would be really slow!
                    545 \def\MT@rem@from@clist#1#2{%
                    546   \def\MT@res@a#1,#1,##2\MT@res@a{##1,##2\MT@res@b}%
                    547   \def\MT@res@b#1,\MT@res@b##2\MT@res@b{\ifx,##1\@empty\else##1\fi}%
                    548   \xdef#2{\MT@exp@two@c\MT@res@b\MT@res@a\expandafter,#2,\MT@res@b,#1,\MT@res@a}%
                    549 }

\MT@in@tlist         Test whether item is in token list. Since this isn't too elegant, I thought that at least
\MT@in@tlist@        here, \pdfmatch would be more efficient – however, it turned out to be even slower
                    than this solution.
                    550 \def\MT@in@tlist#1#2{%
                    551   \MT@inlist@false
                    552   \def\MT@res@a{#1}%
                    553   \MT@map@tlist@c#2\MT@in@tlist@
                    554 }
                    555 \def\MT@in@tlist@#1{%
                    556   \edef\MT@res@b{#1}%
                    557   \ifx\MT@res@a\MT@res@b
                    558     \MT@inlist@true
                    559     \expandafter\MT@tlist@break
                    560   \fi
                    561 }

\MT@in@rlist         Test whether size \MT@size is in a list of ranges. Store the name of the list in
\MT@in@rlist@        \MT@size@name
\MT@in@rlist@@       562 \def\MT@in@rlist#1{%
\MT@size@name        563   \MT@inlist@false
                    564   \MT@map@tlist@c#1\MT@in@rlist@
                    565 }
                    566 \def\MT@in@rlist@#1{\expandafter\MT@in@rlist@@#1}
                    567 \def\MT@in@rlist@@#1#2#3{%
                    568   \MT@ifdim{#2}=\m@ne{%

```

```

569 \MT@ifdim{#1}=\MT@size
570 \MT@inlist@true
571 \relax
572 }{%
573 \MT@ifdim\MT@size<{#1}\relax{%
574 \MT@ifdim\MT@size<{#2}%
575 \MT@inlist@true
576 \relax
577 }%
578 }%
579 \ifMT@inlist@
580 \def\MT@size@name{#3}%
581 \expandafter\MT@tlist@break
582 \fi
583 }

\MT@loop This is the same as LATEX's \loop, which we mustn't use, since this could confuse an
\MT@iterate outer \loop in the document.
\MT@repeat 584 </package>
585 \def\MT@loop#1\MT@repeat{%
586 \def\MT@iterate{#1\relax\expandafter\MT@iterate\fi}%
587 \MT@iterate \let\MT@iterate\relax
588 }
589 \let\MT@repeat\fi

\MT@while@num Execute <#3> from <#1> up to (excluding) <#2> (much faster than LATEX's \@whilenum).
590 \def\MT@while@num#1#2#3{%
591 \@tempcnta#1\relax
592 \MT@loop #3%
593 \advance\@tempcnta \@ne
594 \ifnum\@tempcnta < #2\MT@repeat
595 }

\MT@do@font Execute <#1> 256 times.
596 \def\MT@do@font{\MT@while@num\z@\@cc1vi}
597 <*/package>

\MT@count Increment macro <#1> by one. Saves using up too many counters. The e-TEX way is
\MT@increment slightly faster.
598 \newcount\MT@count
599 \def\MT@increment#1{%
600 ^^X \edef#1{\number\numexpr #1 + 1\relax}%
601 ^^Q \MT@count=#1\relax
602 ^^Q \advance\MT@count \@ne
603 ^^Q \edef#1{\number\MT@count}%
604 }

\MT@scale Multiply and divide a counter. If we are using e-TEX, we will use its \numexpr
primitive. This has the advantage that it is less likely to run into arithmetic overflow.
The result of the division will be rounded instead of truncated. Therefore, we'll get
a different (more accurate) result in about half of the cases.
605 \def\MT@scale#1#2#3{%
606 ^^Q \multiply #1 #2\relax
607 \ifnum #3 = \z@
608 ^^X #1=\numexpr #1 * #2\relax
609 \else
610 ^^X #1=\numexpr #1 * #2 / #3\relax
611 ^^Q \divide #1 #3\relax
612 \fi
613 }

```

Some abbreviations. Thus, we can have short command names but full-length log output.

These we also need the other way round.

We can work on these lists to save some guards in the dtx file.

Whenever an optional argument accepts a list of features, we can use this command to check whether a feature exists in order to prevent a rather confusing ‘Missing \endcsname inserted’ error message. The feature (long form) must be in \@tempa, the type of list to ignore in <#1>, then comes the action.

For the record, the following L^AT_EX kernel commands will be modified by microtype:

- \pickup@font
- \do@subst@correction
- \add@accent (all in section 14.2.9)
- \showwhyphens (in section 14.4.5)

The wordcount package redefines the font-switching commands, which will break microtype. Since microtype doesn't have an effect on the number of words in the document anyway, we will simply disable ourselves.

```
646 \@ifl@aded{tex}{wordcount}{%
```



```

647 \MT@warning@n{Detected the `wordcount' utility.\MessageBreak
648   Disabling `\'MT@MT', since it wouldn't work}%
649 \MT@clearoptions\MT@restore@catcodes\endinput}\relax

\MT@setup@    The setup is deferred until the end of the preamble. This has a couple of advantages:
               \microtypesetup can be used to change options later on in the preamble, and fonts
               don't have to be set up before microtype.

650 /package
651 plain\MT@requires@latex{
652 \let\MT@setup@{}empty

\MT@addto@setup    We use our private hook to have better control over the timing. This will also work
                   with eplain, but not with minilx alone.
653 \def\MT@addto@setup{\g@addto@macro\MT@setup@}

                   Don't hesitate with minilx.
654 plain}{\let\MT@addto@setup@firstofone}

\MT@with@package@T    We almost never do anything if a package is not loaded.
655 \def\MT@with@package@T#1{\ifpackage@loaded{#1}\@firstofone\@gobble}
656 *package

\MT@with@babel@and@T    LATEX's \ifpackage@with ignores the class options.
657 \def\MT@with@babel@and@T#1{%
658   \MT@ifdefined@nT{opt@babel.\@pkgextension}{%
659     \expandtwoargs\MT@inclist{#1}
660     {\csname opt@babel.\@pkgextension\endcsname,\@classoptionslist}%
661     \ifMT@inlist\expandafter\@gobble\fi
662   }\@gobble
663 }

                   Don't load letterspace.
664 \MT@let@nc{ver@letterspace.sty}\@empty

\MT@ledmac@setup    The ledmac package first saves each paragraph in a box, from which it then splits
\MT@led@unhbox@line off the lines one by one. This will destroy character protrusion. (There aren't any
\MT@led@kern         problems with the lineno package, since it takes a different approach.) — ... —
                   After much to and fro, the situation has finally settled and there is a fix. Beginning
                   with pdfTEX version 1.21b together with ledpatch.sty as of 2005/06/02 (v0.4),
                   character protrusion will work at last.

                   Peter Wilson was so kind to provide the \l@dunhbox@line hook in ledmac to
                   allow for protrusion. \leftmarginkern and \rightmarginkern are new primitives
                   of pdfTEX 1.21b (aka. 1.30.0).

665 \MT@requires@pdftex5{
666   \def\MT@ledmac@setup{%
667     \ifMT@protrusion
668       \MT@ifdefined@cTF{\l@dunhbox@line{%
669         \MT@info@n{Patching ledmac to enable character protrusion}%
670         \newdimen\MT@led@kern
671         \let\MT@led@unhbox@line\l@dunhbox@line
672         \renewcommand*{\l@dunhbox@line}[1]{%
673           \ifhbox##1%
674             \MT@led@kern=\rightmarginkern##1%
675             \kern\leftmarginkern##1%
676             \MT@led@unhbox@line##1%
677             \kern\MT@led@kern
678           \fi
679         }%
680       }%

```

```

681      \MT@warning@nl{%
682          Character protrusion in paragraphs with line\MessageBreak
683          numbering will only work if you update ledmac}%
684      }%
685      \fi
686  }
687 }{
688 \def\MT@ledmac@setup{%
689   \ifMT@protrusion
690     \MT@warning@nl{%
691       The pdftex version you are using does not allow\MessageBreak
692       character protrusion in paragraphs with line\MessageBreak
693       numbering by the 'ledmac' package.\MessageBreak
694       Upgrade pdftex to version 1.30 or later}%
695     \fi
696   }
697 }

```

\MT@restore@p@h Restore meaning of \% and \#.

```

698 \def\MT@restore@p@h{\chardef\%\` \% \chardef\#\` \# }

```

\MT@setupfont@hook This hook will be executed every time a font is set up (inside a group).

In the preamble, we check for the packages each time a font is set up. Thus, it will work regardless when the packages are loaded.

Even for packages that don't activate any characters in the preamble (like babel and csquotes), we have to check here, too, in case they were loaded before microtype, and a font is loaded \AtBeginDocument, before microtype. (This is no longer needed, since the complete setup is now deferred until the end of the preamble. However, it is still necessary for defersetup=false.)

```

699 \def\MT@setupfont@hook{%

```

Spanish (and Galician and Mexican) babel modify \%, storing the original meaning in \percentsign.

```

700   \MT@if@false
701   \MT@with@babel@and@T{spanish} \MT@if@true
702   \MT@with@babel@and@T{galician} \MT@if@true
703   \MT@with@babel@and@T{mexican} \MT@if@true
704   \ifMT@if@MT@ifdefined@c@T\percentsign{\let\%\percentsign}\fi

```

Using \@disablequotes, we can restore the original meaning of all characters made active by csquotes. (It would be doable for older versions, too, but we won't bother.)

```

705   \MT@with@package@T{csquotes}{%
706     \@ifpackagelater{csquotes}{2005/05/11}\@disablequotes\relax}%

```

hyperref redefines \% and \# inside a \url. We restore the original meanings (which we can only hope are correct). Same for tex4ht.

```

707   \MT@if@false
708   \MT@with@package@T{hyperref} \MT@if@true
709   \MT@with@package@T{tex4ht} \MT@if@true
710   \ifMT@if@MT@restore@p@h\fi
711 }

```

Check again at the end of the preamble.

```

712 \<package>
713 \MT@addto@setup{%
714 \<package>

```

Our competitor, the pdfcpot package, must not be tolerated!

```

715 \MT@with@package@T{pdfcpot}{%
716 \MT@error{Detected the `pdfcpot' package!\MessageBreak
717 \MT@MT' and `pdfcpot' may not be used together}}{%
718 The `pdfcpot' package provides an interface to character protrusion.\MessageBreak
719 So does the `MT@MT' package. Using both packages at the same\MessageBreak
720 time will almost certainly lead to undesired results. Have your choice!}%
721 }
722 \MT@with@package@T{ledmac}\MT@ledmac@setup

```

We can clean up \MT@setupfont@hook now.

```

723 \let\MT@setupfont@hook\@empty
724 \MT@if@false
725 \MT@with@babel@and@T{spanish} \MT@if@true
726 \MT@with@babel@and@T{galician} \MT@if@true
727 \MT@with@babel@and@T{mexican} \MT@if@true
728 \ifMT@if@
729 \g@addto@macro\MT@setupfont@hook{%
730 \MT@ifdefined@c@T\percentsign{\let%\percentsign}}%
731 \fi
732 \MT@with@package@T{csquotes}{%
733 \ifpackage@later{csquotes}{2005/05/11}}{%
734 \g@addto@macro\MT@setupfont@hook\@disablequotes
735 }{%
736 \MT@warning@nl{%
737 Should you receive warnings about unknown slot\MessageBreak
738 numbers, try upgrading the `csquotes' package}%
739 }%
740 }

```

We disable microtype's additions inside hyperref's \pdfstringdef, which redefines lots of commands. hyperref doesn't work with plain T_EX, so in that case we don't bother.

```

741 \MT@if@false
742 </package>
743 <plain> \MT@requires@latex2{
744 \MT@with@package@T{hyperref}{%
745 \pdfstringdefDisableCommands{%
746 <*package>
747 \let\pickup@font\MT@orig@pickupfont
748 \let\textmicrotypecontext\@secondoftwo
749 \let\microtypecontext\@gobble
750 </package>
751 \def\lststyle{\pdfstringdefWarn\lststyle}%
752 \def\textls#1#\pdfstringdefWarn\textls}%
753 }%
754 <package> \MT@if@true
755 }
756 <plain> }\relax
757 <*package>
758 \MT@with@package@T{tex4ht}\MT@if@true
759 \ifMT@if@\g@addto@macro\MT@setupfont@hook\MT@restore@p@h\fi

```

The listings package makes numbers and letters active,

```

760 \MT@with@package@T{listings}{%
761 \g@addto@macro\MT@cfg@catcodes{%
762 \MT@while@num{"30}{\catcode\@tempcnta 12\relax}%
763 \MT@while@num{"41}{\catcode\@tempcnta 11\relax}%
764 \MT@while@num{"61}{\catcode\@tempcnta 11\relax}%
765 }%

```

... and the backslash (which would lead to problems in \MT@get@slot).

```

766 \g@addto@macro\MT@setupfont@hook{%

```

```
767 \catcode`\z@
```

When loaded with the `extendedchar` option, listings will also redefine 8-bit active characters (`inputenc`). Luckily, this simple redefinition will make them expand to their original definition, so that they could be used in the configuration.

```
768 \let\lst@ProcessLetter\empty
769 }%
770 }
```

Of course, using both `soul`'s and `microtype`'s letterspacing mechanisms at the same time doesn't make much sense. But `soul` can do more, e. g., underlining. The optional argument to `\textls` may not be used.

```
771 </package>
772 <plain> \MT@requires@latex2{
773 \MT@with@package@T{soul}{%
774 \soulregister\lsstyle 0%
775 \soulregister\textls 1%
776 }
```

Under plain $\text{T}_{\text{E}}\text{X}$, `soul` doesn't register itself the $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ way, hence we have to use a different test in this case.

```
777 <*plain>
778 }{\ifx\SOU@ \@undefined\else
779 \soulregister\lsstyle 0%
780 \soulregister\textls 1%
781 \fi}
782 </plain>
783 <*package>
```

Compatibility with the pinyin package (from CJK): disable `microtype` in `\py@macron`, which loads a different font for the accent. In older versions of pinyin (pre-4.6.0), `\py@macron` had only one argument.

```
784 \MT@with@package@T{pinyin}{%
785 \let\MT@orig@py@macron\py@macron
786 \ifpackagelater{pinyin}{2005/08/11}{% 4.6.0
787 \def\py@macron#1#2{%
788 \let\pickup@font\MT@orig@pickupfont
789 \MT@orig@py@macron{#1}{#2}%
790 \let\pickup@font\MT@pickupfont}%
791 }{%
792 \def\py@macron#1{%
793 \let\pickup@font\MT@orig@pickupfont
794 \MT@orig@py@macron{#1}%
795 \let\pickup@font\MT@pickupfont}%
796 }%
797 }
798 </package>
799 }
800 <*package>
```

We need a font (the minimal class doesn't load one).

```
801 \expandafter\ifx\the\font\nullfont\normalfont\fi
```

14.2 Font Setup

`\MT@setupfont` Setting up a font entails checking for each feature whether it should be applied to the current font (`\MT@font`). But first, We might have to disable stuff when used together with adventurous packages.

```
802 \def\MT@setupfont{\MT@setupfont@hook}
```

This will use a copy of the font (allowing for expansion parameter variation and the use of more than one set of protrusion factors for a font within one paragraph).

```
803 \MT@requires@pdftex7
```

```
804 {\g@addto@macro\MT@setupfont\MT@copy@font}\relax
```

The font properties must be extracted from `\MT@font`, since the current value of `\f@encoding` and friends may be wrong!

```
805 \g@addto@macro\MT@setupfont{%
```

```
806 \MT@exp@two@c\MT@split@name\string\MT@font/\@nil
```

Try to find a configuration file for the current font family.

```
807 \MT@exp@one@n\MT@find@file\MT@family
```

```
808 \ifx\MT@familyalias\@empty \else
```

```
809 \MT@exp@one@n\MT@find@file\MT@familyalias\fi
```

We have to make sure that `\cf@encoding` expands to the correct value (for later, in `\MT@get@slot`), which isn't the case when `\selectfont` chooses a new encoding (this would be done a second later in `\selectfont`, anyway – three lines, to be exact). (I think, I do not need this anymore – however, I'm too afraid to remove it. ... Oops, I did it. Let's see whether anybody complains.)

```
810 % \ifx\f@encoding\cf@encoding\else\@enc@update\fi
```

```
811 }
```

Tracking has to come first, since it means actually loading a different font.

```
812 \MT@requires@pdftex6
```

```
813 {\g@addto@macro\MT@setupfont\MT@tracking}\relax
```

```
814 \g@addto@macro\MT@setupfont{%
```

```
815 \MT@check@font
```

```
816 \ifMT@inlist@
```

```
817 (debug)\MT@show@pdfannot2%
```

```
818 \else
```

```
819 \MT@vinfo{Setting up font `~\MT@font'\on@line}%
```

Now we can begin setting up the font for all features that the current pdfTeX provides. The following commands are `\let` to `\relax` if the respective feature is disabled via package options.

For versions older than 1.20, protrusion has to be set up first, beginning with 1.20, the order doesn't matter.

```
820 \MT@protrusion
```

```
821 \MT@expansion
```

```
822 }
```

Interword spacing and kerning (pdfTeX 1.40).

```
823 \MT@requires@pdftex6
```

```
824 {\g@addto@macro\MT@setupfont{\MT@spacing\MT@kerning}}\relax
```

Disable ligatures (pdfTeX 1.30).

```
825 \MT@requires@pdftex5
```

```
826 {\g@addto@macro\MT@setupfont\MT@noligatures}\relax
```

```
827 \g@addto@macro\MT@setupfont{%
```

Debugging.

```
828 (debug)\MT@show@pdfannot1%
```

Finally, register the font so that we don't set it up anew each time.

```
829 \MT@register@font
```

```
830 \fi
```

831 }

\MT@copy@font The new `\pdfcopyfont` command allows to expand a font with different parameters, or to use more than one set of protrusion factors for a given font within one paragraph. It will be used when we find a context for `\SetProtrusion` or `\SetExpansion` in the preamble, or when the package has been loaded with the `copyfonts` option.

```
832 \let\MT@copy@font\relax
833 \MT@requires@pdftex7{
834 \def\MT@copy@font@{%
```

\MT@font@copy For every new protrusion and expansion contexts, we create a new copy.

```
835 \xdef\MT@font@copy{\csname\MT@font/\MT@pr@context/\MT@ex@context\endcsname}%
```

\MT@font@orig pdfTeX doesn't allow to copy a font that has already been copied and expanded/letterspaced. Hence, we have to get the original.

```
836 \expandafter\ifx\MT@font@copy\relax
837 \edef\MT@font@orig{\csname\expandafter\string\font@name @orig\endcsname}%
838 \expandafter\ifx\MT@font@orig\relax
839 \MT@exp@two@c\MT@gl@et\MT@font@orig\font@name
840 \else
841 \MT@exp@two@c\let\font@name\MT@font@orig
842 \fi
843 \global\MT@exp@two@c\pdfcopyfont\MT@font@copy\font@name
844 <debug>\MT@dinfol{creating new copy: \MT@font@copy}%
```

Since it's a new font, we have to remove it from the context lists.

```
845 \MT@map@clist@c\MT@active@features{%
846 \MT@exp@cs\ifx\MT@font@name\MT@abbr@##1}\relax\else
847 \def\@tempa{##1}%
848 \MT@exp@cs\MT@map@tlist@c\MT@##1@doc@contexts\MT@rem@from@list
849 \fi
850 }%
851 \fi
852 \MT@exp@two@c\let\MT@font\MT@font@copy
```

We only need the font identifier for letterspacing.

```
853 \let\font@name\MT@font@copy
```

But we have to properly substitute the font after we're done.

```
854 \aftergroup\let\aftergroup\font@name\aftergroup\MT@font@copy
855 }
```

\MT@rem@from@list

```
856 \def\MT@rem@from@list#1{%
857 \MT@exp@cs\ifx\MT@font@tempa @#1font@list}\relax\else
858 \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
859 \MT@font \csname \MT@font@tempa @#1font@list\endcsname
860 \fi
861 }
862 }\relax
```

Here's the promised dirty trick for users of older pdfTeX versions, which works around the problem that the use of the same font with different expansion parameters is prohibited. If you do not want to create a clone of the font setup (this would require duplicating the `tfm/vf` files under a new name, and writing new `fd` files and map entries), you can load a minimally larger font for the paragraph in question. E. g., for a document typeset in 10 pt:

```
\SetExpansion
```

```
[ stretch = 30,
  shrink  = 60,
  step    = 5 ]
{ encoding = *,
  size = 10.001 }
{ }
\newcommand{\expandpar}[1]{%
  \fontsize{10.001}{\baselineskip}\selectfont #1\par}
% ...
\expandpar{This paragraph contains an 'unnecessary' widow.}
```

Note that the `\expandpar` command can only be applied to complete paragraphs. If you are using Computer Modern Roman, you have to load the `fix-cm` package to be able to select fonts in arbitrary sizes. Finally, the reason I suggest to use a larger font, and not a smaller one, is to prevent a different design size being selected.

```
\MT@split@name    Split up the font name ((#6) may be a protrusion/expansion context and/or a
\MT@encoding      letterspacing amount).
\MT@family 863 \def\MT@split@name#1/#2/#3/#4/#5/#6\@nil{%
864   \def\MT@encoding{#1}%
\MT@series 865   \def\MT@family {#2}%
866   \def\MT@series {#3}%
\MT@shape 867   \def\MT@shape {#4}%
868   \def\MT@size {#5}%

\MT@familyalias   Alias family?
869   \MT@ifdefined@n@TF{MT@MT@family @alias}%
870   {\MT@let@cn\MT@familyalias{MT@MT@family @alias}}%
871   {\let\MT@familyalias\empty}%
872 }

\ifMT@do          We check all features of the current font against the lists of the currently active
\MT@feat          font set, and set \ifMT@do accordingly.
\MT@maybe@do 873 \newif\ifMT@do
874 \def\MT@maybe@do#1{%
    (but only if the feature isn't globally set to false)
875   \csname ifMT@\csname MT@abbr@#1\endcsname\endcsname

    Begin with setting micro-typography to true for this font. The \MT@checklist@...
    tests will set it to false if the property is not in the list. The first non-empty list that
    does not contain a match will stop us (except for font).

876   \MT@dotrue
877   \edef\@tempa{\csname MT@#1@setname\endcsname}%
878   \MT@map@clist@n{font,encoding,family,series,shape,size}{%
879     \MT@ifdefined@n@TF{MT@checklist@#1}%
880     {\csname MT@checklist@#1\endcsname}%
881     {\MT@checklist@{#1}}}%
882   {#1}%
883   }%
884   \else
885     \MT@dofalse
886   \fi
887   \ifMT@do

    \MT@feat stores the current feature.
888   \def\MT@feat{#1}%
889   \csname MT@set@#1@codes\endcsname
890   \else
```

```

891 \MT@vinfo{... No \@nameuse{MT@abbr@#1}}%
892 \fi
893 }

```

\MT@info@list

```

894 <debug>\def\MT@info@list#1#2#3{\MT@info@n1{1}{\@nameuse{MT@abbr@#1}: #2
895 <debug> \ifx\#3\list empty\else \@nameuse{MT@#2}' #3 list\fi}}

```

\MT@checklist@ The generic test (<#1> is the axis, <#2> the feature, \@tempa contains the set name).

```

896 \def\MT@checklist@#1#2{%
897 <!debug> \MT@ifdefined@n@T
898 <debug> \MT@ifdefined@n@TF
899 {MT@#2list@#1@ \@tempa}%

```

Begin an \expandafter orgy to test whether the font attribute is in the list.

```

900 \expandafter\MT@exp@one@n\expandafter\MT@in@clist
901 \csname MT@#1\expandafter\endcsname
902 \csname MT@#2list@#1@ \@tempa\endcsname
903 \ifMT@inlist@
904 <debug>\MT@info@list{#2}{#1}{in}%
905 \MT@dotrue
906 \else
907 <debug>\MT@info@list{#2}{#1}{not in}%
908 \MT@dofalse
909 \expandafter\MT@clist@break
910 \fi
911 }%

```

If no limitations have been specified, i. e., the list for a font attribute has not been defined at all, the font should be set up.

```

912 <debug> {\MT@info@list{#2}{#1}{}}%
913 }

```

\MT@checklist@family Also test for the alias font, if the original font is not in the list.

```

914 \def\MT@checklist@family#1{%
915 <!debug> \MT@ifdefined@n@T
916 <debug> \MT@ifdefined@n@TF
917 {MT@#1list@family@ \@tempa}%
918 \MT@exp@two@n\MT@in@clist
919 \MT@family{\csname MT@#1list@family@ \@tempa\endcsname}%
920 \ifMT@inlist@
921 <debug>\MT@info@list{#1}{family}{in}%
922 \MT@dotrue
923 \else
924 <debug>\MT@info@list{#1}{family}{not in}%
925 \MT@dofalse
926 \ifx\MT@familyalias\empty \else
927 \MT@exp@two@n\MT@in@clist
928 \MT@familyalias{\csname MT@#1list@family@ \@tempa\endcsname}%
929 \ifMT@inlist@
930 <debug> \MT@info@list{#1}{family alias}{in}%
931 \MT@dotrue
932 <debug>\else\MT@info@list{#1}{family alias}{not in}%
933 \fi
934 \fi
935 \fi
936 \ifMT@do \else
937 \expandafter\MT@clist@break
938 \fi
939 }%
940 <debug> {\MT@info@list{#1}{family}{}}%
941 }

```


`\MT@checklist@size` Test whether font size is in list of size ranges.

```

942 \def\MT@checklist@size#1{%
943 <!--debug--> \MT@ifdefined@n@T
944 <!--debug--> \MT@ifdefined@n@TF
945     {MT@#1list@size@%tempa}%
946     \MT@exp@cs\MT@in@rlist{MT@#1list@size@%tempa}%
947     \ifMT@in@rlist@
948 <!--debug-->\MT@info@list{#1}{size}{in}%
949     \MT@dotrue
950   \else
951 <!--debug-->\MT@info@list{#1}{size}{not in}%
952     \MT@dofalse
953     \expandafter\MT@clist@break
954   \fi
955 }%
956 <!--debug--> {\MT@info@list{#1}{size}}}%
957 }

```

`\MT@checklist@font` If the font matches, we skip the rest of the test.

```

958 \def\MT@checklist@font#1{%
959 <!--debug--> \MT@ifdefined@n@T
960 <!--debug--> \MT@ifdefined@n@TF
961     {MT@#1list@font@%tempa}%

```

Since `\MT@font` may be appended with context and/or letterspacing specs, we construct the name from the font characteristics.

```

962     \edef@tempb{\MT@encoding/\MT@family/\MT@series/\MT@shape/\MT@size}%
963     \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter
964     \@tempb \csname MT@#1list@font@%tempa\endcsname
965     \ifMT@in@rlist@
966 <!--debug-->\MT@info@list{#1}{font}{in}%
967     \expandafter\MT@clist@break
968   \else
969 <!--debug-->\MT@info@list{#1}{font}{not in}%
970     \MT@dofalse
971   \fi
972 }%
973 <!--debug--> {\MT@info@list{#1}{font}}}%
974 }

```

14.2.1 Protrusion

`\MT@protrusion` Set up for protrusion?

```

975 \def\MT@protrusion{\MT@maybe@do{pr}}

```

`\MT@set@pr@codes` This macro is called by `\MT@setupfont`, and does all the work for setting up a font for protrusion.

```

976 \def\MT@set@pr@codes{%

```

Check whether and if, which list should be applied to the current font.

```

977   \MT@if@list@exists{%
978     \MT@get@font@dimen@six
979     \MT@get@opt
980     \MT@reset@pr@codes

```

Get the name of the inheritance list and parse it.

```

981   \MT@get@inh@list

```

Set an input encoding?

```

982   \MT@set@inputenc{c}%

```

Load additional lists?

```

983 \MT@load@list\MT@pr@c@name
984 \MT@set@listname

```

Load the main list.

```

985 \MT@let@cn\@tempc{MT@pr@c@\MT@pr@c@name}%
986 \expandafter\MT@set@codes\@tempc,\relax,%
987 }\MT@reset@pr@codes
988 }

```

\MT@gobble@settings

```

989 \def\MT@gobble@settings#1\@tempc,\relax,{}

```

\MT@get@font@dimen@six If \fontdimen 6 is zero, character protrusion won't work, and we can skip the settings (for example, the dsfont fonts don't specify this dimension; this is probably a bug).

```

990 \def\MT@get@font@dimen@six{%
991 \ifnum\fontdimen6\MT@font=\z@
992 \MT@warning@n1{%
993 Font '\MT@font' does not specify its\MessageBreak
994 \backslashchar fontdimen 6 (width of an `em')! Therefore,\MessageBreak
995 \nameuse{MT@abbr@\MT@feat} will not work with this font}%
996 \expandafter\MT@gobble@settings
997 \else
998 \edef\MT@dimen@six{\number\fontdimen6\MT@font}%
999 \fi
1000 }

```

\MT@set@all@pr Set all protrusion codes of the font.

```

1001 \def\MT@set@all@pr#1#2{%
1002 (debug)\MT@info@n1{3}{-- lp/rp: setting all to #1/#2}%
1003 \let\MT@temp\@empty
1004 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\lcode\MT@font\@tempcnta=#1\relax}}%
1005 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\rcode\MT@font\@tempcnta=#2\relax}}%
1006 \MT@do@font\MT@temp
1007 }

```

\MT@reset@pr@codes@ All protrusion codes are zero for new fonts. However, if we have to reload the font due to different contexts, we have to reset them. This command will be changed by \microtypecontext if necessary.

```

1008 \def\MT@reset@pr@codes@{\MT@set@all@pr\z@\z@}
1009 \let\MT@reset@pr@codes\relax

```

\MT@the@pr@code If the font is letterspaced, we have to add half the letterspacing amount to the margin kerns. This will be activated in \MT@set@tr@codes.

```

1010 \def\MT@the@pr@code{\@tempcntb}
1011 \MT@requires@pdftex6{
1012 \def\MT@the@pr@code@tr{%
1013 \numexpr\@tempcntb+\MT@letterspace@/2\relax
1014 }
1015 }\relax

```

\MT@set@codes Split up the values and set the codes.

```

1016 \def\MT@set@codes#1,{%
1017 \ifx\relax#1\@empty\else
1018 \MT@split@codes #1==\relax
1019 \expandafter\MT@set@codes
1020 \fi
1021 }

```

`\MT@split@codes` The `keyval` package would remove spaces here, which we needn't do since `\SetProtrusion` ignores spaces in the protrusion list anyway. `\MT@get@char@unit` may mean different things.

```

1022 \def\MT@split@codes#1=#2=#3\relax{%
1023   \def\@tempa{#1}%
1024   \ifx\@tempa\@empty \else
1025     \MT@get@slot
1026     \ifnum\MT@char > \m@ne
1027       \MT@get@char@unit
1028       \csname MT@\MT@feat @split@val\endcsname#2\relax
1029     \fi
1030   \fi
1031 }

```

`\MT@pr@split@val`

```

1032 \def\MT@pr@split@val#1,#2\relax{%
1033   \def\@tempb{#1}%
1034   \MT@ifempty\@tempb\relax{%
1035     \MT@scale@to@em
1036     \lcode\MT@font\MT@char=\MT@the@pr@code
1037   (debug)\MT@info{n}{4}{;;; lp (\MT@char): \number\lcode\MT@font\MT@char: [#1]}%
1038   }%
1039   \def\@tempb{#2}%
1040   \MT@ifempty\@tempb\relax{%
1041     \MT@scale@to@em
1042     \rcode\MT@font\MT@char=\MT@the@pr@code
1043   (debug)\MT@info{n}{4}{;;; rp (\MT@char): \number\rcode\MT@font\MT@char: [#2]}%
1044   }%

```

Now we can set the values for the inheriting characters. Their slot numbers are saved in the macro `\MT@inh@<list name>@<slot number>@`.

```

1045 \MT@ifdefined@c@T\MT@pr@inh@name{%
1046   \MT@ifdefined@n@T\MT@inh@\MT@pr@inh@name @\MT@char @}{%
1047     \MT@exp@cs\MT@map@tlist@c
1048     {MT@inh@\MT@pr@inh@name @\MT@char @}%
1049     \MT@set@pr@heirs
1050   }%
1051 }%
1052 }

```

`\MT@scale@to@em` Since pdf \TeX version 0.14h, we have to adjust the protrusion factors (i. e., convert numbers from thousandths of character width to thousandths of an em of the font). We have to do this *before* setting the inheriting characters, so that the latter inherit the absolute value, not the relative one if they have a differing width (e. g., the ‘ff’ ligature). Unlike `protcode.tex` and `pdfcprot`, we do not calculate with `\lcode` resp. `\rcode`, since this would disallow protrusion factors larger than the character width (since `\lcode`’s limit is 1000). Now, the maximum protrusion is 1 em of the font.

The unit is in `\MT@count`, the desired factor in `\@tempb`, and the result will be returned in `\@tempcntb`.

```

1053 \MT@requires@pdfTeX3{
1054   \def\MT@scale@to@em{%
1055     \@tempcntb=\MT@count\relax

```

For really huge fonts (100 pt or so), an arithmetic overflow could occur with vanilla \TeX . Using e- \TeX , this can't happen, since the intermediate value is 64 bit, which could only be reached with a character width larger than `\maxdimen`.

```

1056 \MT@scale\@tempcntb \@tempb \MT@dimen@six
1057 \ifnum\@tempcntb=\z@ \else
1058 \MT@scale@factor
1059 \fi
1060 }

```

`\MT@get@charwd` Get the width of the character. When using e-TeX, we can employ `\fontcharwd` instead of building scratch boxes.

```

1061 \def\MT@get@charwd{%
1062 ^^X \MT@count=\fontcharwd\MT@font\MT@char\relax
1063 ^^Q \setbox\z@=\hbox{\MT@font \char\MT@char}%
1064 ^^Q \MT@count=\wd\z@
1065 \ifnum\MT@count=\z@ \MT@info@missing@char \fi
1066 }

```

For letterspaced fonts, we have to subtract the letterspacing amount from the characters' widths. The protrusion amounts will be adjusted in `\MT@set@pr@codes`. The letterspaced font is already loaded so that $1\text{em} = \text{\fontdimen 6}$.

```

1067 \MT@requires@pdfTeX{
1068 \g@addto@macro\MT@get@charwd{%
1069 \MT@ifdefined@c@T\MT@letterspace@
1070 {\advance\MT@count -\dimexpr\MT@letterspace@ sp *\dimexpr 1em/1000\relax}%
1071 }
1072 }\relax
1073 }{

```

No adjustment with versions 0.14f and 0.14g.

```

1074 \def\MT@scale@to@em{%
1075 \MT@count=\@tempb\relax
1076 \ifnum\MT@count=\z@ \else
1077 \MT@scale@factor
1078 \fi
1079 }

```

We need this in `\MT@warn@code@too@large` (neutralised).

```

1080 \def\MT@get@charwd{\MT@count=\MT@dimen@six}
1081 }

```

`\MT@get@font@dimen` For the space unit.

```

1082 \def\MT@get@font@dimen#1{%
1083 \ifnum\fontdimen#1\MT@font=\z@
1084 \MT@warning@n1{Font '\MT@font' does not specify its\MessageBreak
1085 \@backslashchar fontdimen #1 (it's zero)! \MessageBreak
1086 You should use a different 'unit' for \MT@curr@list@name}%
1087 \else
1088 \MT@count=\fontdimen#1\MT@font
1089 \fi
1090 }

```

`\MT@info@missing@char` Info about missing characters, or characters with zero width.

```

1091 \def\MT@info@missing@char{%
1092 \MT@info@n1{Character '\the\MT@toks'
1093 ^^X \iffontchar\MT@font\MT@char
1094 has a width of 0pt
1095 ^^X \else is missing\fi
1096 ^^Q \MessageBreak (it's probably missing)
1097 \MessageBreak in font '\MT@font'. \MessageBreak
1098 Ignoring protrusion settings for this character}%
1099 }

```

`\MT@scale@factor` Furthermore, we might have to multiply with a factor.

```

1100 \def\MT@scale@factor{%
1101   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1102     \expandafter\MT@scale\expandafter \@tempcntb
1103     \csname MT@\MT@feat @factor@\endcsname \@m
1104   \fi
1105   \ifnum\@tempcntb>\csname MT@\MT@feat @max\endcsname\relax
1106     \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @max}%
1107   \else
1108     \ifnum\@tempcntb<\csname MT@\MT@feat @min\endcsname\relax
1109       \MT@exp@cs\MT@warn@code@too@large{MT@\MT@feat @min}%
1110     \fi
1111   \fi
1112 }

```

`\MT@warn@code@too@large` Type out a warning if a chosen protrusion factor is too large after the conversion. As a special service, we also type out the maximum amount that may be specified in the configuration.

```

1113 \def\MT@warn@code@too@large#1{%
1114   \@tempcnta=#1\relax
1115   \ifnum\csname MT@\MT@feat @factor@\endcsname=\@m \else
1116     \expandafter\MT@scale\expandafter\@tempcnta\expandafter
1117     \@m \csname MT@\MT@feat @factor@\endcsname
1118   \fi
1119   \MT@scale\@tempcnta \MT@dimen@six \MT@count
1120   \MT@warning@nl{The \@nameuse{MT@abbr@\MT@feat} code \@tempb\space
1121     is too large for character\MessageBreak
1122     `the\MT@toks' in \MT@curr@list@name.\MessageBreak
1123     Setting it to the maximum of \number\@tempcnta}%
1124   \@tempcntb=#1\relax
1125 }

```

`\MT@get@opt` The optional argument to the configuration commands (except for `\SetExpansion`, which is being dealt with in `\MT@get@ex@opt`).

```

1126 \def\MT@get@opt{%
1127   \MT@set@listname

```

`\MT@pr@factor@` Apply a factor?

```

\MT@sp@factor@ 1128 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}{%
\MT@kn@factor@ 1129 \MT@let@nn{MT@\MT@feat @factor@}
1130 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @factor}%
1131 \MT@vinfo{... : Multiplying \@nameuse{MT@abbr@\MT@feat} codes by
1132 \number\csname MT@\MT@feat @factor@\endcsname/1000}%
1133 }{%
1134 \MT@let@nn{MT@\MT@feat @factor@}{MT@\MT@feat @factor}%
1135 }%

```

`\MT@pr@unit@` The unit can only be evaluated here, since it might be font-specific. If it's `\@empty`, it's relative to character widths, if it's `-1`, relative to space dimensions.

```

\MT@sp@unit@
\MT@kn@unit@ 1136 \MT@ifdefined@n@TF{MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}{%
1137 \MT@let@nn{MT@\MT@feat @unit@}%
1138 {MT@\MT@feat @c@\csname MT@\MT@feat @c@name\endcsname @unit}%
1139 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\@empty
1140 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1141 relative to character widths}%
1142 \else
1143 \MT@exp@cs\ifx{MT@\MT@feat @unit@}\m@ne
1144 \MT@vinfo{... : Setting \@nameuse{MT@abbr@\MT@feat} codes
1145 relative to width of space}%
1146 \fi
1147 \fi
1148 }{%

```

```

1149 \MT@let@nn{MT@MT@feat @unit@}{MT@MT@feat @unit}%
1150 }%

\MT@get@space@unit The codes are either relative to character widths, or to a fixed width. For spacing
\MT@get@char@unit and kerning lists, they may also be relative to the width of the interword glue. Only
the setting from the top list will be taken into account.

1151 \let\MT@get@char@unit\relax
1152 \let\MT@get@space@unit\@gobble
1153 \MT@exp@cs\ifx{MT@MT@feat @unit@}\@empty
1154 \let\MT@get@char@unit\MT@get@charwd
1155 \else
1156 \MT@exp@cs\ifx{MT@MT@feat @unit@}\m@ne
1157 \let\MT@get@space@unit\MT@get@font@dimen
1158 \else
1159 \MT@exp@cs\MT@get@unit{MT@MT@feat @unit@}%
1160 \fi
1161 \fi

Preset all characters? If so, we surely don't need to reset, too.

1162 \MT@ifdefined@n@T{MT@MT@feat @c@\csname MT@MT@feat @c@name\endcsname @preset}{%
1163 \csname MT@preset@MT@feat\endcsname
1164 \MT@let@nc{MT@reset@MT@feat @codes}\relax
1165 }%
1166 }

\MT@get@unit If unit contains an em or ex, we use the corresponding \fontdimen to obtain the
\MT@get@unit@ real size. Simply converting the em into points might give a wrong result, since
the font probably isn't set up yet, so that these dimensions haven't been updated,
either.

1167 \def\MT@get@unit#1{%
1168 \expandafter\MT@get@unit@#1 e!\@nil
1169 \ifx\x\@empty\else\let#1\x\fi
1170 \@defaultunits\@tempdima#1 pt\relax\@nnil
1171 \ifdim\@tempdima=\z@
1172 \MT@warning@n1{%
1173 Cannot set \@nameuse{MT@abbr@MT@feat} factors relative to zero\MessageBreak
1174 width. Setting factors of list \@nameuse{MT@MT@feat @c@name}'\MessageBreak
1175 relative to character widths instead}%
1176 \let#1\@empty
1177 \let\MT@get@char@unit\MT@get@charwd
1178 \else
1179 \MT@vinfo{... : Setting \@nameuse{MT@abbr@MT@feat} factors relative
1180 to \the\@tempdima}%
1181 \MT@count=\@tempdima\relax
1182 \fi
1183 }
1184 \def\MT@get@unit@#1e#2#3\@nil{%
1185 \ifx\#3\\\let\x\@empty \else
1186 \if m#2%
1187 \edef\x{#1\fontdimen6\MT@font}%
1188 \else
1189 \if x#2%
1190 \edef\x{#1\fontdimen5\MT@font}%
1191 \fi
1192 \fi
1193 \fi
1194 }

\MT@set@inputenc The configurations may be under the regime of an input encoding.

1195 \def\MT@set@inputenc#1{%

```

`\MT@cat` We remember the current category (c or inh), in case of warnings later.

```
1196 \def\MT@cat{#1}%
1197 \edef\@tempa{\MT@MT@feat @#1\csname MT@MT@feat @#1\name\endcsname @inputenc}%
1198 \MT@ifdefined@n@T\@tempa\MT@set@inputenc@
1199 }
```

`\MT@set@inputenc@` More recent versions of inputenc remember the current encoding, so that we can test whether we really have to load the encoding file.

```
1200 \MT@addto@setup{%
1201 \ifpackageloaded{inputenc}{
1202 \ifpackageafter{inputenc}{2006/02/22}{
1203 \def\MT@set@inputenc{%
1204 \MT@ifstreq\inputencodingname{\csname\@tempa\endcsname}\relax
1205 \MT@load@inputenc
1206 }
1207 }{
1208 \let\MT@set@inputenc\MT@load@inputenc
1209 }
1210 }{
1211 \def\MT@set@inputenc{%
1212 \MT@warning@n{Key `inputenc' used in \MT@curr@list@name, but the `inputenc'
1213 \MessageBreak package isn't loaded. Ignoring input encoding}%
1214 }
1215 }
1216 }
```

`\MT@load@inputenc` Set up normal catcodes, since, e.g., listings would otherwise want to actually typeset the inputenc file when it is being loaded inside a listing.

```
1217 \def\MT@load@inputenc{%
1218 \MT@cfg@catcodes
1219 (debug)\MT@info@n{1}{loading input encoding: \@nameuse{\@tempa}}%
1220 \inputencoding{\@nameuse{\@tempa}}%
1221 }
```

`\MT@set@pr@heirs` Set the inheriting characters.

```
1222 \def\MT@set@pr@heirs#1{%
1223 \lcode\MT@font#1=\lcode\MT@font\MT@char
1224 \rcode\MT@font#1=\rcode\MT@font\MT@char
1225 (debug)\MT@info@n{2}{-- heir of \MT@char: #1}%
1226 (debug)\MT@info@n{4}{;;; lp/rp (#1): \number\lcode\MT@font\MT@char/%
1227 (debug) \number\rcode\MT@font\MT@char}%
1228 }
```

`\MT@preset@pr` Preset characters. Presetting them relative to their widths is not allowed.

```
\MT@preset@pr@ 1229 \def\MT@preset@pr{%
1230 \expandafter\expandafter\expandafter\MT@preset@pr@
1231 \csname MT@pr@c@\MT@pr@c@name @preset\endcsname\@nil
1232 }
1233 \def\MT@preset@pr@#1,#2\@nil{%
1234 \ifx\MT@pr@unit@\empty
1235 \MT@warn@preset@twidth{pr}%
1236 \let\MT@preset@aux\MT@preset@aux@factor
1237 \else
1238 \def\MT@preset@aux{\MT@preset@aux@space2}%
1239 \fi
1240 \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux{#1}\@tempa}%
1241 \MT@ifempty{#2}{\let\@tempb\@empty}{\MT@preset@aux{#2}\@tempb}%
1242 \MT@set@all@pr\@tempa\@tempb
1243 }
```

`\MT@preset@aux` Auxiliary macro for presetting. Store value `<#1>` in macro `<#2>`.

```

\MT@preset@aux@factor 1244 \def\MT@preset@aux@factor#1#2{%
\MT@preset@aux@space 1245 \@tempcntb=#1\relax
1246 \MT@scale@factor
1247 \edef#2{\number\@tempcntb}%
1248 }
1249 \def\MT@preset@aux@space#1#2#3{%
1250 \def\@tempb{#2}%
1251 \MT@get@space@unit#1%
1252 \MT@scale@to@em
1253 \edef#3{\number\@tempcntb}%
1254 }

\MT@warn@preset@twidth
1255 \def\MT@warn@preset@twidth#1{%
1256 \MT@warning@n1{%
1257 Cannot preset characters relative to their widths\MessageBreak
1258 for \@nameuse{MT@abbr#1} list \@nameuse{MT@#1@c@name}'. Presetting them%
1259 \MessageBreak relative to lem instead}%
1260 }
```

14.2.2 Expansion

`\MT@expansion` Set up for expansion?

```

1261 \def\MT@expansion{\MT@maybe@do{ex}}
```

`\MT@set@ex@codes@` Setting up font expansion is a bit different because of the selected option. There are two versions of this macro.

If `selected=true`, we only apply font expansion to those fonts for which a list has been declared (i. e., like for protrusion).

```

1262 \def\MT@set@ex@codes@{%
1263 \MT@if@list@exists{%
1264 \MT@get@ex@opt
1265 \let\MT@get@char@unit\relax
1266 \MT@reset@ef@codes
1267 \MT@get@inh@list
1268 \MT@set@inputenc{c}%
1269 \MT@load@list\MT@ex@c@name
1270 \MT@set@listname
1271 \MT@let@cn\@tempc{MT@ex@c@\MT@ex@c@name}%
1272 \expandafter\MT@set@codes\@tempc,\relax,%
1273 \MT@expandfont
1274 }\relax
1275 }
```

`\MT@set@ex@codes@n` If, on the other hand, all characters should be expanded by the same amount, we only take the first optional argument to `\SetExpansion` into account.

`\ifMT@nonselected` We need this boolean in `\MT@if@list@exists` so that no warning for missing lists will be issued.

```

1276 \newif\ifMT@nonselected

1277 \def\MT@set@ex@codes@n{%
1278 \MT@nonselectedtrue
1279 \MT@if@list@exists
1280 \MT@get@ex@opt
1281 {%
1282 \let\MT@stretch@ \MT@stretch
1283 \let\MT@shrink@ \MT@shrink
1284 \let\MT@step@ \MT@step
```



```

1285 \let\MT@auto@ \MT@auto
1286 \let\MT@ex@factor@ \MT@ex@factor
1287 }%
1288 \MT@reset@ef@codes
1289 \MT@expandfont
1290 \MT@nonselectedfalse
1291 }

```

`\MT@set@ex@codes` Default is non-selected. It can be changed in the package options.

```
1292 \let\MT@set@ex@codes\MT@set@ex@codes@n
```

`\MT@expandfont` Expand the font.

```

1293 \def\MT@expandfont{%
1294 \pdffontexpand\MT@font \MT@stretch@ \MT@shrink@ \MT@step@ \MT@auto@\relax
1295 }

```

`\MT@set@all@ex` At first, all expansion factors for the characters will be set to 1000 (respectively the
`\MT@reset@ef@codes@` factor of this font).

```

1296 \def\MT@set@all@ex#1{%
1297 (debug)\MT@info{n}{3}{-- ex: setting all to \number#1}%
1298 \MT@do@font{\efcode\MT@font\@tempcnta=#1\relax}%
1299 }
1300 \def\MT@reset@ef@codes@{\MT@set@all@ex\MT@ex@factor@}

```

`\MT@reset@ef@codes` However, this is only necessary for versions prior to 1.20.

```

1301 \MT@requires@pdftex4{
1302 \def\MT@reset@ef@codes{%
1303 \ifnum\MT@ex@factor@=\@m \else
1304 \MT@reset@ef@codes@
1305 \fi
1306 }
1307 }{
1308 \let\MT@reset@ef@codes\MT@reset@ef@codes@
1309 }

```

`\MT@ex@split@val` There's only one number per character.

```

1310 \def\MT@ex@split@val#1\relax{%
1311 \@tempcntb=#1\relax

```

Take an optional factor into account.

```

1312 \ifnum\MT@ex@factor@=\@m \else
1313 \MT@scale\@tempcntb \MT@ex@factor@ \@m
1314 \fi
1315 \ifnum\@tempcntb > \MT@ex@max
1316 \MT@warn@ex@too@large\MT@ex@max
1317 \else
1318 \ifnum\@tempcntb < \MT@ex@min
1319 \MT@warn@ex@too@large\MT@ex@min
1320 \fi
1321 \fi
1322 \efcode\MT@font\MT@char=\@tempcntb
1323 (debug)\MT@info{n}{4}{::: ef (\MT@char): \number\efcode\MT@font\MT@char: [#1]}%

```

Heirs, heirs, I love thy heirs.

```

1324 \MT@ifdefined@c@T\MT@ex@inh@name{%
1325 \MT@ifdefined@n@T{\MT@inh@\MT@ex@inh@name @\MT@char @}{%
1326 \MT@exp@cs\MT@map@tlist@c{\MT@inh@\MT@ex@inh@name @\MT@char @}\MT@set@ex@heirs
1327 }%
1328 }%
1329 }

```

\MT@warn@ex@too@large

```
1330 \def\MT@warn@ex@too@large#1{%
1331   \MT@warning@n1{Expansion factor \number\@tempcntb\space too large for
1332     character\MessageBreak ``the\MT@toks' in \MT@curr@list@name.\MessageBreak
1333     Setting it to the maximum of \number#1}%
1334   \@tempcntb=#1\relax
1335 }
```

\MT@get@ex@opt Apply different values to this font?

```
\MT@ex@factor@ 1336 \def\MT@get@ex@opt{%
\MT@stretch@ 1337   \MT@set@listname
1338   \MT@ifdefined@n@TF{MT@ex@cc@MT@ex@cc@name @factor}{%
\MT@shrink@ 1339     \MT@let@cn\MT@ex@factor@{MT@ex@cc@MT@ex@cc@name @factor}%
\MT@step@ 1340     \MT@vinfo{... : Multiplying expansion factors by \number\MT@ex@factor@/1000}%
\MT@auto@ 1341   }{%
1342     \let\MT@ex@factor@\MT@ex@factor
1343   }%
1344   \MT@get@ex@opt@{stretch}{Setting stretch limit to \number\MT@stretch@}%
1345   \MT@get@ex@opt@{shrink} {Setting shrink limit to \number\MT@shrink@}%
1346   \MT@get@ex@opt@{step}   {Setting expansion step to \number\MT@step@}%
1347   \def\@tempa{autoexpand}%
1348   \MT@get@ex@opt@{auto}{\ifx\@tempa\MT@auto@ En\else Dis\fi abling automatic expansion}%
1349   \MT@ifdefined@n@T{MT@ex@cc@MT@ex@cc@name @preset}{%
1350     \MT@preset@ex
1351     \let\MT@reset@ef@codes\relax
1352   }%
1353 }
```

\MT@get@ex@opt@

```
1354 \def\MT@get@ex@opt@#1#2{%
1355   \MT@ifdefined@n@TF{MT@ex@cc@MT@ex@cc@name @#1}{%
1356     \MT@let@nn{MT@#1@}{MT@ex@cc@MT@ex@cc@name @#1}%
1357     \MT@vinfo{... : #2}%
1358   }{%
1359     \MT@let@nn{MT@#1@}{MT@#1}%
1360   }%
1361 }
```

\MT@set@ex@heirs

```
1362 \def\MT@set@ex@heirs#1{%
1363   \efcode\MT@font#1=\efcode\MT@font\MT@char
1364   (debug)\MT@edinfo@n1{2}{-- heir of \MT@char: #1}%
1365   (debug)\MT@edinfo@n1{4}{::: ef (#1) \number\efcode\MT@font\MT@char}%
1366 }
```

\MT@preset@ex

```
1367 \def\MT@preset@ex{%
1368   \@tempcntb=\csname MT@ex@cc@MT@ex@cc@name @preset\endcsname\relax
1369   \MT@scale@factor
1370   \MT@set@all@ex\@tempcntb
1371 }
```

14.2.3 Interword spacing (glue)

\MT@spacing Adjustment of interword spacing?

```
1372 \MT@requires@pdftex6{
1373   \def\MT@spacing{\MT@maybe@do{sp}}
```

\MT@set@sp@codes This is all the same.

```
1374 \def\MT@set@sp@codes{%
```

```

1375 \MT@if@list@exists{%
1376 \MT@get@font@dimen@six
1377 \MT@get@opt
1378 \MT@reset@sp@codes
1379 \MT@get@inh@list
1380 \MT@set@inputenc{c}%
1381 \MT@load@list\MT@sp@c@name
1382 \MT@set@listname
1383 \MT@let@cn\@tempc\MT@sp@c@\MT@sp@c@name}%
1384 \expandafter\MT@set@codes\@tempc,\relax,%
1385 }\MT@reset@sp@codes
1386 }

```

\MT@sp@split@val If unit=space, \MT@get@space@unit will be defined to fetch the corresponding fontdimen (2 for the first, 3 for the second and 4 for the third argument).

```

1387 \def\MT@sp@split@val#1,#2,#3\relax{%
1388 \def\@tempb{#1}%
1389 \MT@ifempty\@tempb\relax{%
1390 \MT@get@space@unit2%
1391 \MT@scale@to@em
1392 \knbscode\MT@font\MT@char=\@tempcntb
1393 <debug>\MT@info@n1{4}{;;; knbs (\MT@char): \number\knbscode\MT@font\MT@char: [#1]}%
1394 }%
1395 \def\@tempb{#2}%
1396 \MT@ifempty\@tempb\relax{%
1397 \MT@get@space@unit3%
1398 \MT@scale@to@em
1399 \stbscode\MT@font\MT@char=\@tempcntb
1400 <debug>\MT@info@n1{4}{;;; stbs (\MT@char): \number\stbscode\MT@font\MT@char: [#2]}%
1401 }%
1402 \def\@tempb{#3}%
1403 \MT@ifempty\@tempb\relax{%
1404 \MT@get@space@unit4%
1405 \MT@scale@to@em
1406 \shbscode\MT@font\MT@char=\@tempcntb
1407 <debug>\MT@info@n1{4}{;;; shbs (\MT@char): \number\shbscode\MT@font\MT@char: [#3]}%
1408 }%
1409 \MT@ifdefined@c@T\MT@sp@inh@name{%
1410 \MT@ifdefined@n@T\MT@inh@\MT@sp@inh@name @\MT@char @}%
1411 \MT@exp@cs\MT@map@tlist@c\MT@inh@\MT@sp@inh@name @\MT@char @\MT@set@sp@heirs
1412 }%
1413 }%
1414 }

```

\MT@set@sp@heirs

```

1415 \def\MT@set@sp@heirs#1{%
1416 \knbscode\MT@font#1=\knbscode\MT@font\MT@char
1417 \stbscode\MT@font#1=\stbscode\MT@font\MT@char
1418 \shbscode\MT@font#1=\shbscode\MT@font\MT@char
1419 <debug>\MT@info@n1{2}{-- heir of \MT@char: #1}%
1420 <debug>\MT@info@n1{4}{;;; knbs/stbs/shbs (#1): \number\knbscode\MT@font\MT@char/%
1421 <debug> \number\stbscode\MT@font\MT@char/\number\shbscode\MT@font\MT@char}%
1422 }

```

\MT@set@all@sp

```

\MT@reset@sp@codes 1423 \def\MT@set@all@sp#1#2#3{%
\MT@reset@sp@codes@ 1424 <debug>\MT@info@n1{3}{-- knbs/stbs/shbs: setting all to #1/#2/#3}%
1425 \let\MT@temp\empty
1426 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbscode\MT@font\@tempcnta=#1\relax}}%
1427 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\stbscode\MT@font\@tempcnta=#2\relax}}%
1428 \MT@ifempty{#3}\relax{\g@addto@macro\MT@temp{\shbscode\MT@font\@tempcnta=#3\relax}}%
1429 \MT@do@font\MT@temp

```

```

1430 }
1431 \def\MT@reset@sp@codes@\MT@set@all@sp@z@z@z@
1432 \let\MT@reset@sp@codes\relax

\MT@preset@sp
\MT@preset@sp@ 1433 \def\MT@preset@sp{%
1434   \expandafter\expandafter\expandafter\MT@preset@sp@
1435   \csname MT@sp@c@\MT@sp@c@name @preset\endcsname\@nil
1436 }
1437 \def\MT@preset@sp@#1,#2,#3\@nil{%
1438   \ifx\MT@sp@unit@\@empty
1439     \MT@warn@preset@twidth{sp}%
1440     \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@factor{#1}\@tempa}%
1441     \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@factor{#2}\@tempc}%
1442     \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@factor{#3}\@tempb}%
1443   \else
1444     \MT@ifempty{#1}{\let\@tempa\@empty}{\MT@preset@aux@space2{#1}\@tempa}%
1445     \MT@ifempty{#2}{\let\@tempc\@empty}{\MT@preset@aux@space3{#2}\@tempc}%
1446     \MT@ifempty{#3}{\let\@tempb\@empty}{\MT@preset@aux@space4{#3}\@tempb}%
1447   \fi
1448   \MT@set@all@sp@\@tempa\@tempc\@tempb
1449 }
1450 }\relax

```

14.2.4 Additional kerning

\MT@kerning Again, only check for additional kerning for new versions of pdfTeX.

```

1451 \MT@requires@pdftex6{
1452 \def\MT@kerning{\MT@maybe@do{kn}}

```

\MT@set@kn@codes It's getting boring, I know.

```

1453 \def\MT@set@kn@codes{%
1454   \MT@if@list@exists{%
1455     \MT@get@font@dimen@six
1456     \MT@get@opt
1457     \MT@reset@kn@codes
1458     \MT@get@inh@list
1459     \MT@set@inputenc{c}%
1460     \MT@load@list\MT@kn@c@name
1461     \MT@set@listname
1462     \MT@let@cn\@tempc{MT@kn@c@\MT@kn@c@name}%
1463     \expandafter\MT@set@codes\@tempc,\relax,%
1464   }\MT@reset@kn@codes
1465 }

```

\MT@kn@split@val Again, the unit may be measured in the space dimension; this time only \fontdimen 2.

```

1466 \def\MT@kn@split@val#1,#2\relax{%
1467   \def\@tempb{#1}%
1468   \MT@ifempty\@tempb\relax{%
1469     \MT@get@space@unit2%
1470     \MT@scale@to@em
1471     \knbccode\MT@font\MT@char=\@tempcntb
1472 (debug)\MT@edinfo@n1{4}{;;; knbc (\MT@char): \number\knbccode\MT@font\MT@char: [#1]}%
1473   }%
1474   \def\@tempb{#2}%
1475   \MT@ifempty\@tempb\relax{%
1476     \MT@get@space@unit2%
1477     \MT@scale@to@em
1478     \knaccode\MT@font\MT@char=\@tempcntb
1479 (debug)\MT@edinfo@n1{4}{;;; knac (\MT@char): \number\knaccode\MT@font\MT@char: [#2]}%
1480   }%

```

```

1481 \MT@ifdefined@c@T\MT@kn@inh@name{%
1482 \MT@ifdefined@n@T\MT@inh@MT@kn@inh@name @\MT@char @}{%
1483 \MT@exp@cs\MT@map@tlist@c\MT@inh@MT@kn@inh@name @\MT@char @}\MT@set@kn@heirs
1484 }%
1485 }%
1486 }

\MT@set@kn@heirs

1487 \def\MT@set@kn@heirs#1{%
1488 \knbcode\MT@font#1=\knbcode\MT@font\MT@char
1489 \knacode\MT@font#1=\knacode\MT@font\MT@char
1490 (debug)\MT@info@n1{2}{-- heir of \MT@char: #1}%
1491 (debug)\MT@info@n1{4}{;;; knbc (#1): \number\knbcode\MT@font\MT@char/%
1492 (debug) \number\knacode\MT@font\MT@char}%
1493 }

\MT@set@all@kn

\MT@reset@kn@codes 1494 \def\MT@set@all@kn#1#2{%
\MT@reset@kn@codes@ 1495 (debug)\MT@info@n1{3}{-- knac/knbc: setting all to #1/#2}%
1496 \let\MT@temp@empty
1497 \MT@ifempty{#1}\relax{\g@addto@macro\MT@temp{\knbcode\MT@font\@tempcnta=#1\relax}}%
1498 \MT@ifempty{#2}\relax{\g@addto@macro\MT@temp{\knacode\MT@font\@tempcnta=#2\relax}}%
1499 \MT@do@font\MT@temp
1500 }
1501 \def\MT@reset@kn@codes@{\MT@set@all@kn\z@\z@}
1502 \let\MT@reset@kn@codes\relax

\MT@preset@kn

\MT@preset@kn@ 1503 \def\MT@preset@kn{%
1504 \expandafter\expandafter\expandafter\MT@preset@kn@
1505 \csname MT@kn@c@\MT@kn@c@name @preset\endcsname\@nil
1506 }
1507 \def\MT@preset@kn@#1,#2\@nil{%
1508 \ifx\MT@kn@unit@\empty
1509 \MT@warn@preset@towidth{kn}%
1510 \let\MT@preset@aux\MT@preset@aux@factor
1511 \else
1512 \def\MT@preset@aux{\MT@preset@aux@space2}%
1513 \fi
1514 \MT@ifempty{#1}\let\@tempa\@empty{\MT@preset@aux{#1}\@tempa}%
1515 \MT@ifempty{#2}\let\@tempb\@empty{\MT@preset@aux{#2}\@tempb}%
1516 \MT@set@all@kn\@tempa\@tempb
1517 }
1518 }\relax

```

14.2.5 Tracking

This only works with pdfTeX 1.40.

```

1519 \MT@requires@pdftex6{

\MT@tracking      We only check whether a font should not be letterspaced at all, not whether we've
\MT@tracking@     already done that (because we have to do it again).

\MT@tr@font@list 1520 \let\MT@tr@font@list@empty
1521 \def\MT@tracking@{%
1522 \MT@exp@one@n\MT@in@clist\MT@font\MT@tr@font@list
1523 \ifMT@inlist@else
1524 \MT@maybe@do{tr}%
1525 \ifMT@do@else
1526 \xdef\MT@tr@font@list{\MT@tr@font@list\MT@font,}%
1527 \fi
1528 \fi

```

```

1529 }
1530 </package>
1531 \let\MT@tracking
1532 <package> \MT@tracking@
1533 <letterspace> \relax

```

`\MT@set@tr@codes` The tracking amount is determined by the optional argument to `\textls`, settings from `\SetTracking`, or the global `letterspace` option, in this order.

```

1534 \def\MT@set@tr@codes{%
1535 <*package>
1536 \MT@vinfo{Tracking font `\'MT@font'\on@line}%
1537 \MT@get@font@dimen@six
1538 \MT@if@list@exists
1539 \MT@get@tr@opt
1540 \relax
1541 </package>
1542 \MT@ifdefined@c@TF\MT@letterspace@ \relax{\let\MT@letterspace@\MT@letterspace}%
1543 \ifnum\MT@letterspace@=\z@

```

Zero tracking requires special treatment.

```

1544 \MT@set@tr@zero
1545 \else
1546 <package> \MT@vinfo{... Tracking by \number\MT@letterspace@}%

```

Letterspacing only works in PDF mode.

```

1547 \MT@warn@tracking@DVI

```

`\MT@lsfont` The letterspaced font instances are saved in macros `\font name)/letterspacing amount)ls`.

In contrast to `\MT@font`, which may reflect the font characteristics more accurately (taking substitutions into account), `\font@name` is guaranteed to correspond to an actual font identifier.

```

1548 \xdef\MT@lsfont{\csname\expandafter\string\font@name
1549 \number\MT@letterspace@ls\endcsname}%
1550 \expandafter\ifx\MT@lsfont\relax
1551 <debug>\MT@info@nl{1}{... new letterspacing instance}%

```

In case of nested letterspacing with different amounts, we have to extract the base font again.

```

1552 \MT@get@ls@basefont
1553 \global\expandafter\letterspacefont\MT@lsfont\font@name\MT@letterspace@

```

Scale interword spacing (not configurable in `letterspace`).

```

1554 <*package>
1555 \MT@ifdefined@c@TF\MT@tr@ispace
1556 {\let\@tempa\MT@tr@ispace}%
1557 {\edef\@tempa{\MT@letterspace@*,,}}%
1558 \MT@ifdefined@c@TF\MT@tr@ospace
1559 {\edef\@tempa{\@tempa,\MT@tr@ospace}}%
1560 {\edef\@tempa{\@tempa,,,}}%
1561 \expandafter\MT@tr@set@space\@tempa,%
1562 </package>
1563 <*letterspace>
1564 % spacing = {<letterspace amount>*,,}
1565 \fontdimen2\MT@lsfont=\dimexpr\numexpr 1000+\MT@letterspace@ \relax sp
1566 * \fontdimen2\MT@lsfont/1000 \relax
1567 </letterspace>

```

Adjust outer kerning (microtype only).

```

1568 <*package>
1569 \MT@ifdefined@c@TF\MT@tr@okern{\let\@tempa\MT@tr@okern}{\def\@tempa{*,,}}%

```

1570 \expandafter\MT@tr@set@okern\@tempa,%

Disable ligatures (not configurable in letterspace).

1571 \MT@ifdefined@c@T\MT@tr@ligatures\MT@tr@noligatures

1572 </package>

1573 <*letterspace>

1574 % no ligatures = {f}

1575 \tagcode\MT@lsfont`f=\m@ne

1576 </letterspace>

Adjust protrusion values now, and maybe later (in \MT@pr@split@val).

1577 <debug>\MT@info@n1{2}{... compensating for tracking (\number\MT@letterspace@)}%

1578 \MT@do@font{\lcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax

1579 \rcode\MT@lsfont\@tempcnta=\numexpr\MT@letterspace@/2\relax}%

1580 <package> \let\MT@the@pr@code\MT@the@pr@code@tr

1581 \fi

Finally, let the letterspaced font propagate.

1582 \aftergroup\MT@set@lsfont

1583 <package> \let\MT@font\MT@lsfont

\MT@set@curr@ls We need to remember the current letterspacing amount (for \lslig).

\MT@curr@ls 1584 \xdef\MT@set@curr@ls{\def\noexpand\MT@curr@ls{\MT@letterspace@}}%

1585 \aftergroup\MT@set@curr@ls

Adjust surrounding spacing and kerning.

\MT@set@curr@os We get the current outer spacing and adjust it, then, after the end of the current outer group, set the current outer spacing, again, and adjust.

1586 <*package>

1587 \MT@outer@space=\csname MT@outer@space\expandafter\string\font@name\endcsname\relax

1588 \xdef\MT@set@curr@os{\MT@outer@space=\the\MT@outer@space\relax}%

1589 \aftergroup\aftergroup\aftergroup\MT@set@curr@os

1590 \MT@tr@outer@l

1591 </package>

If \MT@ls@adjust is empty, it's the starred version of \textls. Use scaling to avoid a 'Dimension too large'.

1592 \ifx\MT@ls@adjust\@empty

1593 <letterspace> % \textls : outer kerning = {*,*} ; \textls* : outer kerning = {0,0}

1594 \MT@outer@kern=-\dimexpr\MT@letterspace@ sp * \fontdimen6\font@name/2000\relax

1595 \MT@ls@outer@k

1596 <*letterspace>

1597 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%

1598 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok

1599 \aftergroup\aftergroup\aftergroup\MT@ls@outer@k

1600 </letterspace>

Otherwise, get the current outer kerning and adjust it, for left and right side (microtype only).

1601 <*package>

1602 \else

1603 \MT@outer@kern=\expandafter\expandafter\expandafter\@firstoftwo

1604 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax

1605 \ifdim\MT@outer@kern=\z@ \else \MT@ls@outer@k \fi

1606 \MT@outer@kern=\expandafter\expandafter\expandafter\@secondoftwo

1607 \csname MT@outer@kern\expandafter\string\font@name\endcsname\relax

1608 </package>

1609 \fi

1610 <*package>

\MT@set@curr@ok Carry the outer kerning amount to outside the next group, then set outer spacing

(which will set kerning, if no space follows).

```

1611 \xdef\MT@set@curr@ok{\MT@outer@kern=\the\MT@outer@kern\relax}%
1612 \aftergroup\aftergroup\aftergroup\MT@set@curr@ok
1613 \aftergroup\aftergroup\aftergroup\MT@tr@outer@r
1614 </package>
1615 \fi
1616 }

```

\MT@get@tr@opt Various settings (only for the microtype version).

```

1617 <*package>
1618 \def\MT@get@tr@opt{%
1619 \MT@set@listname
1620 \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name}{%
1621 \MT@let@cn\MT@letterspace{MT@tr@c@\MT@tr@c@name}%

```

\MT@tr@unit@ Different unit?

```

1622 \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @unit}{%
1623 \MT@let@cn\MT@tr@unit@{MT@tr@c@\MT@tr@c@name @unit}%
1624 \ifdim\MT@tr@unit@=1em
1625 \let\MT@tr@unit@ \@undefined
1626 \else
1627 \MT@let@cn\@tempb{MT@tr@c@\MT@tr@c@name}%
1628 \MT@get@unit\MT@tr@unit@
1629 \let\MT@tr@factor@\@m
1630 \MT@scale@to@em
1631 \edef\MT@letterspace{\number\@tempcntb}%
1632 \fi
1633 }%
1634 }%

```

\MT@tr@ispace Adjust interword spacing.

```

\MT@tr@ospace 1635 \MT@get@tr@opt@{spacing} {ispace}%
1636 \MT@get@tr@opt@{outerspacing}{ospace}%

```

\MT@tr@okern Adjust outer kerning.

```

1637 \MT@get@tr@opt@{outerkerning}{okern}%

```

\MT@tr@ligatures Which ligatures should we disable (empty means all, undefined none)?

```

1638 \MT@get@tr@opt@{noligatures} {ligatures}%
1639 }

```

\MT@get@tr@opt@

```

1640 \def\MT@get@tr@opt@#1#2{%
1641 \MT@ifdefined@n@T{MT@tr@c@\MT@tr@c@name @#1}%
1642 {\MT@let@nn{MT@tr@#2}{MT@tr@c@\MT@tr@c@name @#1}}%
1643 }
1644 </package>

```

\MT@set@lsfont Redefine \font@name, which will be called a second later (in \selectfont).

```

1645 <plain>\MT@requires@latex2{
1646 \def\MT@set@lsfont{\MT@exp@two@c\let\font@name\MT@lsfont}

```

\lsstyle Disable the tests whether the font should be letterspaced, then trigger the setup. Only \textls can be used in math mode (\lsstyle may be used inside another text switch, of course).

```

1647 \DeclareRobustCommand\lsstyle{%
1648 \not@math@alphabet\lsstyle\textls
1649 <package> \def\MT@feat{tr}%
1650 \let\MT@tracking\MT@set@tr@codes
1651 \selectfont

```


1652 }

Now the definitions for the `letterspace` package with plain \TeX .

```

1653 <plain>
1654 {}{
1655 \def\MT@set@lsfont{\MT@lsfont}
1656 \def\lsstyle{%
1657   \begingroup
1658   \escapechar\m@ne
1659   \xdef\font@name{\csname\expandafter\string\the\font\endcsname}%
1660   \MT@set@tr@codes
1661   \endgroup
1662 }
1663 \let\textls\undefined
1664 \let\lslig\undefined
1665 }
1666 </plain>

```

`\lslig` For Fraktur fonts, some ligatures shouldn't be broken up. This command will temporarily select the base font and insert the correct kerning.

```

1667 \DeclareRobustCommand\lslig[1]{%
1668   {\MT@ifdefined@c@TF\MT@curr@ls{%
1669     \escapechar\m@ne
1670     \MT@get@ls@basefont
1671     \MT@outer@kern=\dimexpr\MT@curr@ls sp * \fontdimen6\font@name/2000\relax
1672     \kern\MT@outer@kern
1673     \font@name #1%
1674     \kern\MT@outer@kern%
1675   }{#1}}}%
1676 }

```

`\MT@ls@basefont` pdf \TeX cannot letterspace fonts that already are letterspaced. Therefore, we have to save the base font in `\font name)@base`.

The previous solution (checking the macro's meaning with `\pdfmatch`), where we were loading the base font via the `\font` primitive again, would destroy all previously set up micro-typographic features of the font.

```

1677 \def\MT@get@ls@basefont{%
1678   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1679   \expandafter\ifx\MT@ls@basefont\relax
1680     \MT@exp@two@c\MT@gl@et\MT@ls@basefont\font@name
1681   \else
1682     <debug>\MT@info@n{1}{... fixing base font}%
1683     \MT@exp@two@c\let\font@name\MT@ls@basefont
1684   \fi
1685 }

```

`\MT@set@ls@basefont` If tracking is switched off in the middle of the document, or if `\textls` is called with a zero letterspacing amount, we have to retrieve the base font and select it.

```

1686 \def\MT@set@ls@basefont{\MT@exp@two@c\let\font@name\MT@ls@basefont}
1687 \def\MT@set@tr@zero{%
1688   <debug>\MT@info@n{1}{... zero tracking}%
1689   \xdef\MT@ls@basefont{\csname\expandafter\string\font@name @base\endcsname}%
1690   \expandafter\ifx\MT@ls@basefont\relax \else
1691     <debug>\MT@info@n{1}{... fixing base font}%
1692     \aftergroup\MT@set@ls@basefont
1693   \fi
1694 }

```

`\MT@tr@noligatures` pdf \TeX 1.40.0–1.40.3 disabled all ligatures in letterspaced fonts.

```

1695 <package>
1696 \MT@requirespdfTeX{

```

```

1697 \def\MT@tr@noligatures{%
1698   \ifx\MT@tr@ligatures\@empty
1699     \MT@noligatures@\MT@lsfont\@undefined
1700   \else
1701     \MT@noligatures@\MT@lsfont\MT@tr@ligatures
1702   \fi
1703 }
1704 }{
1705 \def\MT@tr@noligatures{%
1706   \MT@warning@n1{%
1707     Disabling selected ligatures is only possible since\MessageBreak
1708     pdftex 1.40.4. Disabling all ligatures instead}%
1709   \MT@gl@et\MT@tr@noligatures\relax
1710 }
1711 }

```

\MT@outer@space A new skip for outer spacing.

```
1712 \newskip\MT@outer@space
```

\MT@tr@set@space Adjust interword spacing (\fontdimen 2–4) for inner and outer space. For inner spacing, the font dimensions will be adjusted, the settings for outer spacing will be remembered in a macro.

```

1713 \def\MT@tr@set@space#1,#2,#3,#4,#5,#6,{%
1714 <debug>\MT@din@n12{... orig. space: \the\fontdimen2\MT@lsfont,
1715 <debug>   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont
1716 <debug>   \MessageBreak... (#1,#2,#3) (#4,#5,#6)}%
1717   \let\MT@temp\@empty
1718   \MT@tr@set@space@{#1}{#4}{2}\@empty
1719   \MT@tr@set@space@{#2}{#5}{3}\@plus
1720   \MT@tr@set@space@{#3}{#6}{4}\@minus
1721   \MT@gl@et@nc{\MT@outer@space\expandafter\string\font@name}\MT@temp
1722 <debug>\MT@din@n12{... inner space: \the\fontdimen2\MT@lsfont,
1723 <debug>   \the\fontdimen3\MT@lsfont, \the\fontdimen4\MT@lsfont}%
1724 <debug>\MT@din@n12{... outer space: \MT@temp}%
1725 }

```

\MT@tr@set@space@ If outer spacing settings don't exist, they will be inherited from the inner spacing settings.

```

1726 \def\MT@tr@set@space@#1#2#3#4{%
1727   \MT@ifempty{#2}{%
1728     \MT@ifempty{#1}{%
1729       \edef\MT@temp{\MT@temp#4\the\fontdimen#3\MT@lsfont}%
1730     }{%
1731       \MT@tr@set@space@@{#1}{#3}{1000}%
1732       \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1733       \fontdimen#3\MT@lsfont=\@tempdima
1734     }%
1735   }{%
1736     \MT@tr@set@space@@{#2}{#3}{2000}%
1737     \edef\MT@temp{\MT@temp#4\the\@tempdima}%
1738     \MT@ifempty{#1}\relax{%
1739       \MT@tr@set@space@@{#1}{#3}{1000}%
1740       \fontdimen#3\MT@lsfont=\@tempdima
1741     }%
1742   }%
1743 }

```

\MT@tr@set@space@@ If the value is followed by an asterisk, the fontdimen will be scaled by the respective amount, otherwise the value denotes the desired dimension in the respective unit.

```

1744 \def\MT@tr@set@space@@#1#2#3{%
1745   \MT@test@ast#1*\@nil{%

```

```

1746 \MT@ifdefined@ec@TF\MT@tr@unit@
1747 {\edef\@tempb{#1}\MT@scale@to@em}
1748 {\@tempcntb=#1\relax}%
1749 \@tempdima=\dimexpr \dimexpr\@tempcntb sp*\MT@dimen@six/1000\relax
1750 -\fontdimen#2\MT@lsfont\relax

```

For `\fontdimen 2`, we also have to subtract the kerning that letterspacing adds to the sides of the characters (only half if it's for outer spacing).

```

1751 \ifnum#2=\tw@
1752 \advance\@tempdima -\dimexpr\MT@letterspace@ sp*\MT@dimen@six/#3\relax
1753 \fi
1754 \@tempdima=\dimexpr \fontdimen#2\MT@lsfont+\@tempdima\relax
1755 }{%
1756 \MT@ifempty\@tempa{\let\@tempa\MT@letterspace@}\relax
1757 \@tempdima=\dimexpr \numexpr1000+\@tempa sp *\fontdimen#2\MT@lsfont/1000\relax
1758 }%
1759 <debug>\MT@info@n13{... : font dimen #2 (#1): \the\@tempdima}%
1760 }

```

`\MT@tr@outer@l` Recall the last skip (must really be an interword space, not just a marker, nor a ‘hard’ space, i. e., one that doesn’t contain stretch or shrink parts).

```

1761 \def\MT@tr@outer@l{%
1762 \ifhmode
1763 \ifdim\lastskip>5sp
1764 \edef\x{\the\lastskip minus 0pt}%
1765 \setbox\z@\hbox{\MT@outer@space=\x}%
1766 \ifdim\wd\z@>\z@
1767 <debug>\MT@info@2{[[[ adjusting pre space: \the\MT@outer@space}%
1768 \unskip \hskip\MT@outer@space\relax

```

Disable left outer kerning.

```

1769 \let\MT@ls@outer@k\relax
1770 \else

```

The `ragged2e` package sets `\spaceskip` without glue.

```

1771 \ifdim\lastskip=%
1772 \ifnum\spacefactor<2000
1773 \spaceskip
1774 \else
1775 \ifdim\xspaceskip=\z@
1776 \dimexpr\spaceskip+\fontdimen7\font@name\relax
1777 \else
1778 \xspaceskip
1779 \fi
1780 \fi
1781 <debug>\MT@info@2{[[[ adjusting pre space (skip): \the\MT@outer@space}%
1782 \unskip \hskip\MT@outer@space\relax
1783 \let\MT@ls@outer@k\relax
1784 \fi
1785 \fi
1786 \fi
1787 \fi
1788 }

```

`\MT@tr@outer@next` The following is borrowed from `soul`. I’ve added the cases for italic correction, since tracking may also be triggered by text commands (e. g., `\textsc`).

```

\MT@tr@outer@r
\MT@tr@outer@r@ 1789 \def\MT@tr@outer@r@{%
1790 \futurelet\MT@tr@outer@next\MT@tr@outer@r@
1791 }
1792 \def\MT@tr@outer@r@{%

```

Don’t adjust in math mode. There was a tricky bug when `\textls` was the last

command in a `\mathchoice` group.

```
1793 \ifmode
1794   \def\MT@temp*{%
1795   \else
```

A similar bug occurred when adjustment would happen inside a discretionary group, which we prevent here. This only works with e-TeX (which we know is available).

```
1796   \ifnum\currentgrouptype=10 \else
1797   \def\MT@temp*##1{\ifhmode\hskip\MT@outer@space
1798 (debug)\MT@edinfo2{}} adjusting post spaces (1): \the\MT@outer@space}%
1799   \fi}%
1800   \ifcat\egroup\noexpand\MT@tr@outer@next

1801   \ifhmode\unkern\fi\egroup
1802   \MT@set@curr@ok
1803   \MT@set@curr@os
1804   \def\MT@temp*{\afterassignment\MT@tr@outer@r\let\MT@temp=}%
1805   \else
```

If the next token is `\maybe@ic` (from an enclosing text command), we gobble it, read the next one, feed it to `\maybe@ic@` (via `\MT@tr@outer@icr`) and then call ourselves again.

```
1806   \ifx\maybe@ic\MT@tr@outer@next
1807   \def\MT@temp*{\afterassignment\MT@tr@outer@icr\let\MT@temp=}%
1808   \else
```

If the next token is `\check@icr` (from an inner text command), we insert ourselves just before it. This will then call `\maybe@ic` again the next round (which however will always insert an italic correction, since it doesn't read beyond our group).

```
1809   \ifx\check@icr\MT@tr@outer@next
1810   \def\MT@temp*{\aftergroup\MT@tr@outer@r\check@icr\let\MT@temp=}%
1811   \else
1812   \ifx\@sptoken\MT@tr@outer@next
1813   \def\MT@temp* {\ifhmode\hskip\MT@outer@space
1814 (debug)\MT@edinfo2{}} adjusting post spaces (2): \the\MT@outer@space}%
1815   \fi}%
1816   \else
1817   \ifx-\MT@tr@outer@next
1818   \def\MT@temp*~{\nobreak\hskip\MT@outer@space
1819 (debug)\MT@edinfo2{}} adjusting post spaces (3): \the\MT@outer@space}%
1820   }%
1821   \else
1822   \ifx\ \MT@tr@outer@next \else
1823   \ifx\space\MT@tr@outer@next \else
1824   \ifx\@xobeysp\MT@tr@outer@next \else
```

If there's no outer spacing, there may be outer kerning.

```
1825   \def\MT@temp*{\ifdim\MT@outer@kern=\z@\else\MT@ls@outer@k\fi}%
1826   \let\MT@tr@outer@next\relax
1827   \fi\fi\fi\fi\fi\fi\fi\fi\fi
1828   \MT@temp*%
1829 }
```

`\MT@tr@outer@icr` Helper macros for the italic correction mess.

```
\MT@tr@outer@icr@ 1830 \def\MT@tr@outer@icr{\afterassignment\MT@tr@outer@icr@\MT@tr@outer@r}
1831 \def\MT@tr@outer@icr@{%
1832   \let\@let@token= \MT@tr@outer@next
1833   \maybe@ic@
1834 }
```

For older pdf \TeX versions, throw an error.

```

1835 }{
1836   \DeclareRobustCommand\lsstyle{%
1837     \MT@error{Letterspacing only works with pdftex version 1.40\MessageBreak
1838       or newer}{Upgrade pdftex, or use the `soul' package instead.}%
1839     \MT@glet\lsstyle\relax
1840   }
1841 }

```

And for lua \TeX , too.

```

1842 (*lua)
1843 \MT@requires@luatex{
1844   \DeclareRobustCommand\lsstyle{%
1845     \MT@error{Letterspacing currently doesn't work with luatex}
1846       {Run pdftex, or use the `soul' package instead.}%
1847     \MT@glet\lsstyle\relax
1848   }
1849 } \relax
1850 (/lua)
1851 (/package)

```

`\textls` This command may be used like the other text commands. The starred version removes kerning on the sides. The optional argument changes the letterspacing factor.

```

1852 \DeclareRobustCommand\textls{%
1853   \@ifstar{\let\MT@ls@adjust@empty\MT@textls}%
1854   {\let\MT@ls@adjust@empty\MT@relax}%
1855 }

```

`\MT@textls` This is now almost L^A \TeX 's `\DeclareTextFontCommand`, with the difference that we adjust the outer spacing and kerning also for `\lsstyle`, while L^A \TeX 's text *switches* don't bother about italic correction.

```

1856 \newcommand\MT@textls[2][]{%
1857   \ifmmode
1858     \nfss@text{\MT@ls@set@ls{#1}\lsstyle#2}%
1859   \else
1860     \hmode\bgroup
1861       \MT@ls@set@ls{#1}%
1862       \lsstyle #2%
1863       \expandafter
1864       \egroup
1865     \fi
1866 }

```

`\MT@ls@adjust` Set current letterspacing amount and outer kerning. This has to be done inside the same group as the letterspacing command.

```

\MT@ls@adjust@empty
\MT@ls@adjust@relax
\MT@ls@set@ls
1867 \def\MT@ls@adjust@empty{\let\MT@ls@adjust@empty}
1868 \def\MT@ls@adjust@relax{\let\MT@ls@adjust@relax}
1869 \def\MT@ls@set@ls#1{%
1870   \MT@ifempty{#1}%
1871   {\let\MT@letterspace@undefined}%
1872   {\KV@spdef\MT@letterspace@{#1}%
1873     \MT@ls@too@large\MT@letterspace@}%
1874   \MT@ls@adjust@
1875 }

```

`\MT@ls@too@large` Test whether letterspacing amount is too large.

```

1876 \def\MT@ls@too@large#1{%
1877   \ifnum#1>\MT@tr@max
1878     \MT@warning{Maximum for option `letterspace' is \number\MT@tr@max}%
1879     \let#1\MT@tr@max

```

```

1880 \else
1881 \ifnum#1<\MT@tr@min
1882 \MT@warning{Minimum for option `letterspace' is \number\MT@tr@min}%
1883 \let#1\MT@tr@min
1884 \fi
1885 \fi
1886 }

\MT@outer@kern This dimen is used for the starred version of \textls, for \lslig and for adjusted
\MT@tr@set@okern outer kerning.

1887 \newdimen\MT@outer@kern
1888 <package>
1889 \def\MT@tr@set@okern#1,#2,{%
1890 \let\MT@temp@empty
1891 \MT@ifempty{#1}{\MT@tr@set@okern@{}}{\MT@tr@set@okern@{#1}}%
1892 \MT@ifempty{#2}{\MT@tr@set@okern@{}}{\MT@tr@set@okern@{#2}}%
1893 \MT@gl@et@nc{\MT@outer@kern\expandafter\string\font@name}\MT@temp
1894 <debug>\MT@info@n12{... outer kerning: (#1,#2)
1895 <debug> = \@nameuse{\MT@outer@kern\expandafter\string\font@name}}%
1896 }

\MT@tr@set@okern@
1897 \def\MT@tr@set@okern@#1{%
1898 \MT@test@ast#1*\@nil{%
1899 \MT@ifdefined@c@TF\MT@tr@unit@
1900 {\edef\@tempb{#1}\MT@scale@to@em}
1901 {\@tempcntb=#1\relax}%
1902 \@tempdima=\dimexpr \@tempcntb sp * \MT@dimen@six/1000\relax
1903 }%
1904 \MT@ifempty\@tempa{\let\@tempa\@m}\relax
1905 \@tempdima=\dimexpr \numexpr\@tempa*\MT@letterspace@/1000\relax sp
1906 * \fontdimen6\MT@lsfont/2000\relax
1907 }%
1908 \advance\@tempdima -\dimexpr \MT@letterspace@ sp
1909 * \fontdimen6\MT@lsfont/2000\relax
1910 \edef\MT@temp{\MT@temp{\the\@tempdima}}%
1911 }
1912 </package>

\MT@ls@outer@k Adjust outer kerning.
1913 \def\MT@ls@outer@k{\ifhmode\kern\MT@outer@kern\relax\fi}
1914 <package>

```

14.2.6 Disabling ligatures

`\MT@noligatures` The possibility to disable ligatures is a new features of pdf \TeX 1.30.

```

1915 \MT@requires@pdftex5{
1916 \def\MT@noligatures{%
1917 \MT@dotrue
1918 \let\@tempa\MT@n1@setname
1919 \MT@map@clist@n{font,encoding,family,series,shape,size}{%
1920 \MT@ifdefined@n@TF{\MT@checklist@##1}%
1921 {\csname MT@checklist@##1\endcsname}%
1922 {\MT@checklist@##1}}%
1923 {n1}%
1924 }%
1925 \ifMT@do
1926 \MT@noligatures@\MT@font\MT@n1@ligatures
1927 \fi
1928 }

```

`\MT@noligatures@` This is also used by `\MT@set@tr@codes`.

```
1929 \def\MT@noligatures@#1#2{%
1930   \MT@ifdefined@c@TF#2{%
```

Early MiKTeX versions (before 2.5.2579) didn't know `\tagcode`.

```
1931   \MT@ifdefined@c@TF\tagcode{%
```

No 'inputenc' key.

```
1932     \let\MT@warn@maybe@inputenc\@empty
1933     \def\MT@curr@list@name{\@backslashchar DisableLigatures}%
1934     \MT@map@c@list@c#2{%
1935       \KV@sp@def\@tempa{##1}\MT@get@slot
1936       \ifnum\MT@char>\m@ne \tagcode#1\MT@char=\m@ne \fi}%
1937     \MT@vinfo{... Disabling ligatures for characters: #2}%
1938   }{%
1939     \pdfnoligatures#1%
1940     \MT@warning{Cannot disable selected ligatures (pdftex doesn't\MessageBreak
1941       know \@backslashchar tagcode). Disabling all ligatures of\MessageBreak
1942       the font instead}%
1943   }%
1944 }{%
1945   \pdfnoligatures#1%
1946   \MT@vinfo{... Disabling ligatures}%
1947 }%
1948 }
1949 }\relax
```

14.2.7 Loading the configuration

`\MT@load@list` Recurse through the lists to be loaded.

```
1950 \def\MT@load@list#1{%
1951   \edef\@tempa{#1}%
1952   \MT@let@cn\@tempb{MT@\MT@feat @c@\@tempa @load}%
1953   \MT@ifstreq\@tempa\@tempb{%
1954     \MT@error{\@nameuse{MT@abbr@\MT@feat} list '\@tempa' cannot load itself}{}%
1955   }{%
1956     \ifx\@tempb\relax \else
1957       \MT@ifdefined@n@TF{MT@\MT@feat @c@\@tempb}{%
1958         \MT@vinfo{... : First loading \@nameuse{MT@abbr@\MT@feat} list '\@tempb'}%
1959         \begin@group
1960           \MT@load@list\@tempb
1961         \end@group
1962         \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list
1963           \noexpand\MessageBreak'\@tempb'}%
1964         \MT@let@cn\@tempc{MT@\MT@feat @c@\@tempb}%
1965         \expandafter\MT@set@codes\@tempc,\relax,%
1966       }{%
1967         \MT@error{\@nameuse{MT@abbr@\MT@feat} list '\@tempb' undefined.\MessageBreak
1968           Cannot load it from list '\@tempa'}{}%
1969       }%
1970     \fi
1971   }%
1972 }
```

`\MT@find@file` Micro-typographic settings may be written into a file `mt-(font family).cfg`.

`\MT@file@list` We must also record whether we've already loaded the file.

```
1973 \let\MT@file@list\@empty
1974 \def\MT@find@file#1{%
```

Check for existence of the file only once.

```

1975 \MT@in@clist{#1}\MT@file@list
1976 \ifMT@inlist@ \else

```

Don't forget that because reading the files takes place inside a group, all commands that may be used there have to be defined globally.

```

1977 \MT@begin@catcodes
1978 \let\MT@begin@catcodes\relax
1979 \let\MT@end@catcodes\relax
1980 \InputIfFileExists{mt-#1.cfg}{%
1981 \edef\MT@curr@file{mt-#1.cfg}%
1982 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1983 \MT@xadd\MT@file@list{#1,}%
1984 }{%
1985 \MT@get@basefamily#1\@empty\@empty\@empty\@nil
1986 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
1987 \ifMT@inlist@
1988 \MT@xadd\MT@file@list{#1,}%
1989 \else
1990 \InputIfFileExists{mt-\@tempa.cfg}{%
1991 \edef\MT@curr@file{mt-\@tempa.cfg}%
1992 \MT@vinfo{... Loading configuration file \MT@curr@file}%
1993 \MT@xadd\MT@file@list{\@tempa,#1,}%
1994 }{%
1995 \MT@vinfo{... No configuration file mt-#1.cfg}%
1996 \MT@xadd\MT@file@list{#1,}%
1997 }%
1998 \fi
1999 }%
2000 \endgroup
2001 \fi
2002 }

```

`\MT@cfg@catcodes` We have to make sure that all characters have the correct category code. Especially, new lines and spaces should be ignored, since files might be loaded in the middle of the document. This is basically `\nfss@catcodes` (from the \LaTeX kernel). I've added: & (in tabulars), !, ?, , , \$, -, ~, and = (Turkish babel).

OK, now all printable characters up to 127 are 'other'. We hope that letters are always letters and numbers other. (`\listings` makes them active, see section 14.1.5.)

We leave ^ at catcode 7, so that stuff like '^ff' remains possible.

```

2003 \def\MT@cfg@catcodes{%
2004 \makeatletter
2005 \catcode\^7%
2006 \catcode\ 9%
2007 \catcode\^^I9%
2008 \catcode\^^M9%
2009 \catcode\\\z@
2010 \catcode\{\@ne
2011 \catcode\}\@tw@
2012 \catcode\#6%
2013 \catcode\%14%
2014 \MT@map@tlist@n
2015 {\!\"\$&'(\)\*+,\-\.\/\:\;\<=\>\?[\]\_-\|\~}%
2016 \makeother
2017 }

```

`\MT@begin@catcodes` This will be used before reading the files as well as in the configuration commands `\Set...`, and `\DeclareCharacterInheritance`, so that the catcodes are also harmless when these commands are used outside the configuration files.

```

2018 \def\MT@begin@catcodes{%
2019 \begingroup

```


Table 4: Order for matching font attributes

	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Encoding	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Family	•	•	•	•	•	•	•	•	-	-	-	-	-	-	-	-
Series	•	•	•	•	-	-	-	-	•	•	•	•	-	-	-	-
Shape	•	•	-	-	•	•	-	-	•	•	-	-	•	•	-	-
Size	•	-	•	-	•	-	•	-	•	-	•	-	•	-	•	-

```
2020 \MT@cfg@catcodes
2021 }
```

\MT@end@catcodes End group if outside configuration file (otherwise relax).

```
2022 \let\MT@end@catcodes\endgroup
```

\MT@get@basefamily The family name might have a suffix e. g., for expert set (x), old style numbers (j) swash capitals (w) etc. We mustn't simply remove the last letter, as this would make for instance cms out of cmss and cmsy (OK, cmex will still become cme ...).

We only work on the font name if it is longer than three characters.

```
2023 \def\MT@get@basefamily#1#2#3#4\@nil{%
2024 \ifx\@empty#4%
2025 \def\@tempa{#1#2#3}%
2026 \else
2027 \let\@tempa\@empty
2028 \edef\@tempb{#1#2#3#4}%
2029 \expandafter\MT@get@basefamily@\@tempb\@nil
2030 \fi
2031 }
```

\MT@get@basefamily@ This will only remove one suffix (the longest match), so that combinations of suffixes would have to be added manually (e. g., \DeclareMicrotypeVariants*{aw}). But otherwise, something like 'padx' would be truncated to 'p'.

```
2032 \def\MT@get@basefamily@#1#2\@nil{%
2033 \edef\@tempa{\@tempa#1}%
2034 \ifx\#2\@expandafter\@gobble\else\expandafter\@firstofone\fi
2035 {\MT@in@tlist{#2}\MT@variants
2036 \ifMT@inlist\else\MT@get@basefamily@#2\@nil\fi}%
2037 }
```

\MT@listname Try all combinations of font family, series, shape and size to get a list for the current font.

```
\MT@get@listname
\MT@get@listname@ 2038 \def\MT@get@listname#1{%
2039 (debug)\MT@info{n1}{trying to find \@nameuse{MT@abbr@#1} list for font '\MT@font'}%
2040 \let\MT@listname\undefined
2041 \def\@tempb{#1}%
2042 \MT@map@tlist@c\MT@try@order\MT@get@listname@
2043 }
2044 \def\MT@get@listname@#1{%
2045 \expandafter\MT@next@listname#1%
2046 \ifx\MT@listname\undefined \else
2047 \expandafter\MT@tlist@break
2048 \fi
2049 }
```

\MT@try@order Beginning with version 1.7, we always check for the font size. Since the matching order has become more logical now, it can be described in words, so that we don't need table 4 in the documentation part any longer and can cast it off here.

```
2050 \def\MT@try@order{%
```

```

2051 {1111}{1110}{1101}{1100}{1011}{1010}{1001}{1000}%
2052 {0111}{0110}{0101}{0100}{0011}{0010}{0001}{0000}%
2053 }

```

`\MT@next@listname` The current context is added to the font attributes. That is, the context must match.

```

2054 \def\MT@next@listname#1#2#3#4{%
2055   \edef\@tempa{\MT@encoding
2056     /\ifnum#1=\@ne \MT@family\fi
2057     /\ifnum#2=\@ne \MT@series\fi
2058     /\ifnum#3=\@ne \MT@shape\fi
2059     /\ifnum#4=\@ne *\fi
2060     \MT@context}%
2061   (debug)\MT@info{n1}{1}{trying \@tempa}%
2062   \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2063     \MT@next@listname@#4%
2064   }{%

```

Also try with an alias family.

```

2065   \ifnum#1=\@ne
2066     \ifx\MT@familyalias\@empty \else
2067       \edef\@tempa{\MT@encoding
2068         /\MT@familyalias
2069         /\ifnum#2=\@ne \MT@series\fi
2070         /\ifnum#3=\@ne \MT@shape\fi
2071         /\ifnum#4=\@ne *\fi
2072         \MT@context}%
2073     (debug)\MT@info{n1}{1}{(alias) \@tempa}%
2074     \MT@ifdefined@n@TF{MT@\@tempb @\@tempa}{%
2075       \MT@next@listname@#4%
2076     }%
2077   \fi
2078 \fi
2079 }%
2080 }

```

`\MT@next@listname@` If size is to be evaluated, do that, otherwise use the current list.

```

2081 \def\MT@next@listname@#1{%
2082   \ifnum#1=\@ne
2083     \MT@exp@cs\MT@in@rlist{MT@\@tempb @\@tempa @sizes}%
2084     \ifMT@inlist@
2085       \let\MT@listname\MT@size@name
2086     \fi
2087   \else
2088     \MT@let@cn\MT@listname{MT@\@tempb @\@tempa}%
2089   \fi
2090 }

```

`\MT@if@list@exists`

```

\MT@context 2091 \def\MT@if@list@exists{%
2092   \MT@let@cn\MT@context{MT@\MT@feat @context}%
2093   \MT@ifstreq{@}\MT@context{\let\MT@context\@empty}\relax
2094   \MT@get@listname{\MT@feat @c}%
2095   \MT@ifdefined@c@TF\MT@listname{%
2096     \MT@edef@n{MT@\MT@feat @c@name}{\MT@listname}%
2097     \ifMT@nonselected
2098       \MT@vinfo{... Applying non-selected expansion (list '\MT@listname')}%
2099     \else
2100       \MT@vinfo{... Loading \@nameuse{MT@abbr@\MT@feat} list '\MT@listname'}%
2101     \fi
2102     \@firstoftwo
2103   }{%

```

Since the name cannot be \@empty, this is a sound proof that no matching list exists.

```
2104 \MT@let@enc{MT@MT@feat @c@name}\@empty
```

Don't warn if selected=false.

```
2105 \ifMT@nonselected
2106 \MT@vinfo{... Applying non-selected expansion (no list)}%
2107 \else
```

Tracking doesn't require a list, either.

```
2108 \MT@ifstreq\MT@feat{tr}\relax{%
2109 \MT@warning{I cannot find a \@nameuse{MT@abbr@MT@feat} list
2110 for font\MessageBreak`MT@font'%
2111 \ifx\MT@context\@empty\else\space(context: `MT@context')\fi.
2112 Switching off\MessageBreak\@nameuse{MT@abbr@MT@feat} for this font}%
2113 }%
2114 \fi
2115 \@secondoftwo
2116 }%
2117 }
```

\MT@get@inh@list The inheritance lists are global (no context).

```
\MT@context 2118 \def\MT@get@inh@list{%
2119 \let\MT@context\@empty
2120 \MT@get@listname{\MT@feat @inh}%
2121 \MT@ifdefined@c@TF\MT@listname{%
2122 \MT@edef@n{MT@MT@feat @inh@name}{\MT@listname}%
2123 <debug>\MT@info@n{1}{... Using \@nameuse{MT@abbr@MT@feat} inheritance list
2124 <debug> \MT@listname'}%
2125 \MT@let@cn@tempc{MT@MT@feat @inh@MT@listname}%
```

If the list is \@empty, it has already been parsed.

```
2126 \ifx\@tempc\@empty \else
2127 <debug>\MT@info@n{1}{parsing inheritance list ...}%
```

The group is only required in case an input encoding is given.

```
2128 \begingroup
2129 \edef\MT@curr@list@name{inheritance list\noexpand\MessageBreak`MT@listname'}%
2130 \MT@set@inputenc{inh}%
2131 \expandafter\MT@inh@do\@tempc,\relax,%
2132 \MT@gl@et@nc{MT@MT@feat @inh@MT@listname}\@empty
2133 \endgroup
2134 \fi
2135 }%
2136 \MT@let@enc{MT@MT@feat @inh@name}\@undefined
2137 }%
2138 }
```

14.2.8 Translating characters into slots

Get the slot number of the character in the current encoding.

\MT@get@slot There are lots of possibilities how a character may be specified in the configuration files, which makes translating them into slot numbers quite expensive. Also, we want to have this as robust as possible, so that the user does not have to solve a sphinx's riddle if anything goes wrong.

\MT@char The character is in \@tempa, we want its slot number in \MT@char.

```
\MT@char@ 2139 \def\MT@get@slot{%
2140 \escapechar`\
```

```

2141 \let\MT@char@m@ne
2142 \MT@noresttrue

```

Save unexpanded string in case we need to issue a warning message.

```
2143 \MT@toks=\expandafter{\@tempa}%
```

Now, let's walk through (hopefully) all possible cases.

- It's a letter, a character or a number.

```

2144 \expandafter\MT@is@letter\@tempa\relax\relax
2145 \ifnum\MT@char@ < \z@

```

- It might be an active character, i. e., an 8-bit character defined by `inputenc`. If so, we will expand it here to its LICR form.

```
2146 \MT@exp@two@c\MT@is@active\string\@tempa\@nil
```

- OK, so it must be a macro. We do not allow random commands but only those defined in \LaTeX 's idiosyncratic font encoding scheme:

If $\langle encoding \rangle \langle command \rangle$ (that's *one* command) is defined, we try to extract the slot number.

We must be cautious not to stumble over accented characters consisting of two commands, like `\'i` or `\U\CYRI`, hence, `\string` wouldn't be safe enough.

```

2147 \MT@ifdefined@nTF{\MT@encoding\MT@detokenizec\@tempa}%
2148 \MT@is@symbol

```

- Now, we'll catch the rest, which hopefully is an accented character (e. g. `\'a`).

```

2149 {\expandafter\MT@is@composite\@tempa\relax\relax}%
2150 \ifnum\MT@char@ < \z@

```

- It could also be a `\chardefed` command (e. g., the percent character). This seems the least likely case, so it's last.

```

2151 \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2152 \meaning\expandafter\@tempa\MT@charstring\relax\relax\relax
2153 \fi
2154 \fi

```

```

2155 \let\MT@char\MT@char@
2156 \ifnum\MT@char < \z@
2157 \MT@warn@unknown
2158 \else

```

If the user has specified something like `'fi`', or wanted to define a number but forgot to use three digits, we'll have something left of the string. In this case, we issue a warning and forget the complete string.

```

2159 \ifMT@norest \else
2160 \MT@warn@rest
2161 \let\MT@char@m@ne
2162 \fi
2163 \fi
2164 \escapechar@m@ne
2165 }

```

`\ifMT@norest` Test whether all of the string has been used up.

```
2166 \newif\ifMT@norest
```

`\MT@is@letter` Input is a letter, a character or a number.

```

2167 \def\MT@is@letter#1#2\relax{%
2168   \ifcat a\noexpand#1\relax
2169     \edef\MT@char@{\number`#1}%
2170     \ifx\#2\%
2171 <debug>\MT@info@n1{3}{> `the\MT@toks' is a letter (\MT@char@)}%
2172     \else
2173       \MT@norestfalse
2174     \fi
2175   \else
2176     \ifcat !\noexpand#1\relax
2177       \edef\MT@char@{\number`#1}%
2178 <debug>\MT@info@n1{3}{> `the\MT@toks' is a character (\MT@char@)}%
2179       \ifx\#2\%
2180         \ifnum\MT@char@ > 127 \MT@warn@ascii \fi
2181       \else
2182         \MT@norestfalse
2183         \expandafter\MT@is@number#1#2\relax\relax
2184       \fi
2185     \fi
2186   \fi
2187 }
```

`\MT@is@number` Numbers may be specified as a three-digit decimal number (029), as a hexadecimal number (prefixed with "": "1D) or as a octal number (prefixed with ': '35). They must consist of at least three characters (including the prefix), that is, "F is not permitted.

```

2188 \def\MT@is@number#1#2#3\relax{%
2189   \ifx\relax#3\relax \else
2190     \ifx\relax#2\relax \else
2191       \MT@noresttrue
2192       \if#1"\relax
2193         \def\x{\uppercase{\edef\MT@char@{\number#1#2#3}}}\x
2194 <debug>\MT@info@n1{3}{> ... a hexadecimal number: \MT@char@}%
2195       \else
2196         \if#1'\relax
2197         \def\MT@char@{\number#1#2#3}%
2198 <debug>\MT@info@n1{3}{> ... an octal number: \MT@char@}%
2199       \else
2200         \MT@ifint{#1#2#3}{%
2201           \def\MT@char@{\number#1#2#3}%
2202 <debug>\MT@info@n1{3}{> ... a decimal number: \MT@char@}%
2203         }\MT@norestfalse
2204       \fi
2205     \fi
2206     \ifnum\MT@char@ > \@cclv
2207       \MT@warn@number@too@large{\noexpand#1\noexpand#2\noexpand#3}%
2208       \let\MT@char@\m@ne
2209     \fi
2210   \fi
2211 \fi
2212 }
```

`\MT@is@active` Expand an active character. (This was completely broken in v1.7, and only worked by chance before.) We `\set@display@protect` to translate, e. g., `Ä` into `\"A`, that is to whatever it is defined in the `inputenc` encoding file.

Unfortunately, the (older) `inputenc` definitions prefer the protected/generic variants (e. g., `\copyright` instead of `\textcopyright`), which our parser won't be able to understand. (I'm fed up now, so you have to complain if you really,

really want to be able to write ‘©’ instead of \textcopyright, thus rendering your configuration files unportable.)

Unicode characters (inputenc/utf8,utf8x) are also supported.

```
2213 \def\MT@is@active#1#2\@nil{%
2214   \ifnum\catcode`#1 = \active
2215     \begingroup
2216     \set@display@protect
2217     \let\IeC\@firstofone
2218     \let\@inpenc@undefined\MT@undefined@char
```

We refrain from checking whether there is a sufficient number of octets.

```
2219   \def\UTFviii@defined##1{\ifx ##1\relax
2220     \MT@undefined@char{utf8}\else\expandafter ##1\fi}%
```

For ucs (utf8x). Let’s call it experimental ...

```
2221   \MT@ifdefined@cT\PrerenderUnicode
2222   {\PrerenderUnicode{\@tempa}\let\unicode@charfilter\@firstofone}%
2223   \edef\x{\endgroup
2224     \def\noexpand\@tempa{\@tempa}%
```

Append what we think the translation is to the token register we use for the log.

```
2225   \MT@toks={\the\MT@toks\space(= \@tempa)}%
2226   }%
2227   \x
2228   \fi
2229 }
```

\MT@undefined@char For characters not defined in the current input encoding.

```
2230 \def\MT@undefined@char#1{undefined in input encoding ``#1'')}
```

\MT@is@symbol The symbol commands might expand to funny stuff, depending on context. Instead of simply expanding \<command>, we construct the command \<encoding>\<command> and see whether its meaning is \char" <hex number>, which is the case for everything that has been defined with \DeclareTextSymbol in the encoding definition files.

```
2231 \def\MT@is@symbol{%
2232   \expandafter\def\expandafter\MT@char\expandafter
2233   {\csname\MT@encoding\MT@detokenize@c\@tempa\endcsname}%
2234   \expandafter\MT@exp@two@c\expandafter\MT@is@char\expandafter
2235   \meaning\expandafter\MT@char\MT@charstring\relax\relax\relax
2236   \ifnum\MT@char@ < \z@
```

... or, if it hasn’t been defined by \DeclareTextSymbol, a letter (e. g., \i, when using frenchpro).

```
2237   \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2238   \fi
2239 }
```

\MT@is@char A helper macro that inspects the \meaning of its argument.

```
\MT@charstring 2240 \begingroup
2241   \catcode`\=/\z@
2242   /MT@map@tlist@n{/CHAR}/@makeoother
2243   /lowercase{%
2244     /def/x{/endgroup
2245     /def/MT@charstring{\CHAR}%
2246     /def/MT@is@char##1\CHAR"##2##3##4/relax{%
2247       /ifx/relax##1/relax
2248       /if##3\relax
2249       /edef/MT@char@{/number"##2}%
2250       /MT@ifstreq/MT@charstring{##3##4}/relax/MT@noestfalse
```

```

2251         /else
2252         /edef/MT@char@{/number"##2##3}%
2253         /MT@ifstreq/MT@charstring{##4}/relax/MT@noestfalse
2254         /fi
2255 (debug) /MT@dinfo@n1{3}{> `~/the/MT@toks' is a \char (/MT@char@)}%
2256         /fi
2257     }%
2258 }%
2259 }
2260 /x

```

`\MT@is@composite` Here, we are dealing with accented characters, specified as two tokens.

```

2261 \def\MT@is@composite#1#2\relax{%
2262     \ifx\#2\\\else

```

Again, we construct a control sequence, this time of the form: `\\(encoding)(accent)-(character)`, e. g., `\\T1"-a`, which we then expand once to see if it is a letter (if it has been defined by `\DeclareTextComposite`). This should be robust, finally, especially, since we also `\detokenize` the input instead of only `\stringifying` it. Thus, we will die gracefully even on wrong Unicode input without `utf8`.

```

2263     \expandafter\def\expandafter\MT@char\expandafter{\csname\expandafter
2264         \string\csname\MT@encoding\endcsname
2265         \MT@detokenize@n{#1}-\MT@detokenize@n{#2}\endcsname}%
2266     \expandafter\expandafter\expandafter\MT@is@letter\MT@char\relax\relax
2267     \fi
2268 }

```

[What about math? Well, for a moment the following looked like a solution, with `\mt@is@mathchar` defined accordingly, analogous to `\MT@is@char` above, to pick up the last two tokens (the `\meaning` of a `\mathchardef`'ed command expands to its hexadecimal notation):

```

\def\MT@is@mathchar#1{%
  \if\relax\noexpand#1% it's a macro
    \let\x#1%
  \else % it's a character
    \mathchardef\x=\mathcode~#1\relax
  \fi
  \expandafter\MT@exp@two@c\expandafter\mt@is@mathchar\expandafter
  \meaning\expandafter\x\mt@mathcharstring\relax\relax\relax
}

```

However, the problem is that `\mathcodes` and `\mathchardefs` have global scope. Therefore, if they are changed by a package that loads different math fonts, there is no guarantee whatsoever that things will still be correct (e. g., the minus in `cmsy` when the `euler` package is loaded). So, no way to go, unfortunately.]

Some warning messages, for performance reasons separated here.

`\MT@curr@list@name` The type and name of the current list, defined at various places.

```

\MT@set@listname 2269 \def\MT@set@listname{%
2270     \edef\MT@curr@list@name{\@nameuse{MT@abbr@\MT@feat} list\noexpand\MessageBreak
2271     \@nameuse{MT@\MT@feat @c@name}}}%
2272 }

```

`\MT@warn@ascii` For 'other' characters > 127, we issue a warning (`inputenc` probably hasn't been loaded), since correspondence with the slot numbers would be purely coincidental.

```

2273 \def\MT@warn@ascii{%
2274     \MT@warning@n1{Character `~/the/MT@toks' (= \MT@char@)
2275     is outside of ASCII range.\MessageBreak

```

```

2276     You must load the \inputenc package before using \MessageBreak
2277     8-bit characters in \MT@curr@list@name%
2278 }

\MT@warn@number@too@large    Number too large.
2279 \def\MT@warn@number@too@large#1{%
2280   \MT@warning@nl{%
2281     Number #1 in encoding \MT@encoding too large!\MessageBreak
2282     Ignoring it in \MT@curr@list@name%
2283 }

\MT@warn@rest    Not all of the string has been parsed.
2284 \def\MT@warn@rest{%
2285   \MT@warning@nl{%
2286     Unknown slot number of character\MessageBreak`the\MT@toks'
2287     \MT@warn@maybe@inputenc\MessageBreak
2288     in font encoding \MT@encoding.\MessageBreak
2289     Make sure it's a single character\MessageBreak
2290     (or a number) in \MT@curr@list@name%
2291 }

\MT@warn@unknown    No idea what went wrong.
2292 \def\MT@warn@unknown{%
2293   \MT@warning@nl{%
2294     Unknown slot number of character\MessageBreak`the\MT@toks'
2295     \MT@warn@maybe@inputenc\MessageBreak
2296     in font encoding \MT@encoding in \MT@curr@list@name%
2297 }

\MT@warn@maybe@inputenc    In case an input encoding had been requested.
2298 \def\MT@warn@maybe@inputenc{%
2299   \MT@ifdefined@n@T
2300   { \MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc}%
2301   { (input encoding \@nameuse
2302     { \MT@MT@feat @\MT@cat @\csname MT@MT@feat @\MT@cat @name\endcsname @inputenc'})}%
2303 }

```

14.2.9 Hook into L^AT_EX's font selection

We append `\MT@setupfont` to `\pickup@font`, which is called by L^AT_EX every time a font is selected. We then check whether we've already seen this font, and if not, set it up for micro-typography. This ensures that we will catch all fonts, and that we will not set up fonts more than once. The whole package really hangs on this command.

In contrast to the `pdfcprot` package, it is not necessary to declare in advance which fonts should benefit from micro-typographic treatment. Also, only those fonts that are actually being used will be set up.

For my reference:

- `\pickup@font` is called by `\selectfont`, `\wrong@fontshape`, or `\getanddefine@fonts` (for math).
- `\pickup@font` calls `\define@newfont`.
- `\define@newfont` may call (inside a group!)
 - `\wrong@fontshape`, which in turn will call `\pickup@font`, and thus `\define@newfont` again, or

– \extract@font.

- \get@external@font is called by \extract@font, by itself, and by the substitution macros.

Up to version 1.3 of this package, we were using \define@newfont as the hook, which is only called for *new* fonts, and therefore seemed the natural choice. However, this meant that we had to take special care to catch all fonts: we additionally had to set up the default font, the error font (if it wasn't the default font), we had to check for some packages that might have been loaded before microtype and were loading fonts, e. g., jurabib, ledmac, pi font (loaded by hyperref), tipa, and probably many more. Furthermore, we had to include a hack for the IEEEtran class which loads all fonts in the class file itself (to fine tune inter-word spacing), and the memoir class, too. To cut this short: it seemed to get out of hand, and I decided that it would be better to use \pickup@font and decide for ourselves whether we've already seen that font. I hope the overhead isn't too large.

\MT@font@list We use a comma separated list.

```
\MT@font 2304 \let\MT@font@list\empty
2305 \let\MT@font\empty
```

All this is done at the beginning of the document. It doesn't work for plain, of course, which doesn't have \pickup@font.

```
2306 </package>
2307 <plain>\MT@requires@latex2{
2308 \MT@addto@setup{%
```

\MT@orig@pickupfont microtype also works with CJK in the sense that nothing will break when both packages are used at the same time. However, since CJK has its own way of encoding, it is currently not possible to create character-specific settings. That is, the only feature available with CJK fonts is expansion. (Tracking doesn't really work for other reasons.) Like us, CJK redefines \pickup@font.

```
2309 \ifpackageloaded{CJK}{
2310   \ifpackageafter{CJK}{2006/10/17} % 4.7.0
2311   {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}
2312    {\def\MT@orig@pickupfont{\CJK@ifundefined{CJK@plane}}}}
2313   \g@addto@macro\MT@orig@pickupfont
2314   {\expandafter\ifx\font@name\relax\define@newfont\fi}}
```

CJKutf8 redefines \pickup@font once more (recent versions, in PDF mode, as determined by ifpdf, which CJKutf8 loads).

```
2315 \ifpackageloaded{CJKutf8}
2316   {\ifpackageafter{CJKutf8}{2008/05/22} % 4.8.0
2317    {\ifpdf\expandafter\secondoftwo\else\expandafter\firstoftwo\fi}
2318    {\@firstoftwo}}
2319   {\@firstoftwo}
2320   {\g@addto@macro\MT@orig@pickupfont{%
2321    {\expandafter\ifx\csname\curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2322     \define@newfont\else\xdef\font@name{%
2323      \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}}
2324   {\g@addto@macro\MT@orig@pickupfont{%
2325    {\expandafter\ifx\csname \curr@fontshape/\f@size/\CJK@plane\endcsname\relax
2326     \define@newfont\def\CJK@temp{v}%
2327     \ifx\CJK@temp\CJK@plane
2328       \expandafter\ifx\csname CJK@cmaph\@family\CJK@plane\endcsname\relax
2329       \else\csname CJK@cmaph\@family\CJK@plane\endcsname\fi
2330       \else \CJK@addcmaph\CJK@plane \fi
```

```

2331     \else\edef\font@name{%
2332         \csname \curr@fontshape/\f@size/\CJK@plane\endcsname}\fi}}
2333     }{
2334     \def\MT@orig@pickupfont{\expandafter\ifx\font@name\relax\define@newfont\fi}
2335     }

```

Check whether `\pickup@font` is defined as expected. The warning issued by `\CheckCommand*` would be a bit too generic.

```

2336     \ifx\pickup@font\MT@orig@pickupfont \else
2337     \MT@warning@n1{%
2338         Command \string\pickup@font\space is not defined as expected.%
2339         \MessageBreak Patching it anyway. Some things may break%
2340     }*package
2341     .\MessageBreak Double-check whether micro-typography is indeed%
2342     \MessageBreak applied to the document.%
2343     \MessageBreak (Hint: Turn on `verbose' mode)%
2344 }/package
2345     }
2346     \fi

```

`\pickup@font` Then we append our stuff. Everything is done inside a group.

```

2347     \g@addto@macro\pickup@font{\begingroup}

```

If the `trace` package is loaded, we turn off tracing of microtype's setup, which is extremely noisy.

```

2348     \MT@with@package@T{trace}{\g@addto@macro\pickup@font{\conditionally@traceoff}}
2349     \g@addto@macro\pickup@font{%
2350         \escapechar\m@ne
2351     }*package
2352     debug \global\MT@inannottrue
2353     debug \MT@gl@et\MT@pdf@annot\@empty
2354     debug \MT@addto@annot{(line \number\inputlineno)}%

```

If `\MT@font` is empty, no substitution has taken place, hence `\font@name` is correct. Otherwise, if they are different, `\font@name` does not describe the font actually used. This test will catch first order substitutions, like `bx` to `b`, but it will still fail if the substituting font is itself substituted.

```

2355     \MT@let@cn\MT@font\MT@subst\expandafter\string\font@name}%
2356     \ifx\MT@font\relax
2357     \let\MT@font\font@name
2358     \else
2359     \ifx\MT@font\font@name \else
2360     debug \MT@addto@annot{= substituted with \MT@font}%
2361     \MT@register@subst@font
2362     \fi
2363     \fi
2364     \MT@setupfont
2365 }/package
2366 letterspace \MT@tracking
2367     \endgroup
2368     }
2369 }*package

```

`\MT@pickupfont` Remember the patched command for later.

```

2370     \let\MT@pickupfont\pickup@font

```

`\do@subst@correction` Additionally, we hook into `\do@subst@correction`, which is called if a substitution has taken place, to record the name of the ersatz font. Unfortunately, this will only work for one-level substitutions. We have to remember the substitute for the rest of the document, not just for the first time it is called, since we need it every time a

font is letterspaced.

```
2371 \g@addto@macro\do@subst@correction
2372 {\edef\MT@font{\csname\curr@fontshape/\f@size\endcsname}%
2373 \MT@gl@et@nc{MT@subst@expandafter\string\font@name}\MT@font}
```

\add@accent Inside \add@accent, we have to disable microtype’s setup, since the grouping in
\MT@orig@add@accent the patched \pickup@font would break the accent if different fonts are used for
the base character and the accent. Fortunately, L^AT_EX takes care that the fonts used
for the \accent are already set up, so that we cannot be overlooking them.

```
2374 \let\MT@orig@add@accent\add@accent
2375 \def\add@accent#1#2{%
2376 \let\pickup@font\MT@orig@pickupfont
2377 \MT@orig@add@accent{#1}{#2}%
2378 \let\pickup@font\MT@pickupfont
2379 }
2380 </package>
2381 }
2382 <plain>\relax
2383 <*package>
```

Consequently (if all goes well), we are the last ones to change these commands, therefore there is no need to check whether our definition has survived.

\MT@check@font Check whether we’ve already seen the current font.

```
2384 \def\MT@check@font{\MT@exp@one@n\MT@in@clist\MT@font\MT@font@list}
```

\MT@register@subst@font Register the substituted font.

```
2385 \def\MT@register@subst@font{\xdef\MT@font@list{\MT@font@list\font@name,}}
```

\MT@register@font Register the current font.

```
2386 \def\MT@register@font{\xdef\MT@font@list{\MT@font@list\MT@font,}}
```

14.2.10 Context-sensitive setup

Here are the variants for context-sensitive setup.

\MT@active@features The activated features are stored in this command.

```
2387 \let\MT@active@features\@empty
```

\MT@check@font@cx Every feature has its own list of fonts that have already been dealt with. If the
font needn’t be set up for a feature, we temporarily disable the corresponding
setup command. This should be more efficient than book-keeping the fonts in lists
associated with the combination of contexts, as we’ve done it before.

```
2388 \def\MT@check@font@cx{%
2389 \MT@if@true
2390 \MT@map@clist@c\MT@active@features{%
2391 \expandafter\MT@exp@one@n\expandafter\MT@in@clist\expandafter\MT@font
2392 \csname MT@##1@\csname MT@##1@context\endcsname font@list\endcsname
2393 \ifMT@inlist@
2394 \MT@let@nc{MT@\@nameuse{MT@abbr@##1}}\relax
2395 \else
2396 \MT@if@false
2397 \fi
2398 }%
2399 \ifMT@if@ \MT@inlist@true \else \MT@inlist@false \fi
2400 }
```

\MT@register@subst@font@cx Add the substituted font to each feature list.

```
2401 \def\MT@register@subst@font@cx{%
```

```

2402 \MT@map@clist@c\MT@active@features{%
2403   \MT@exp@cs\MT@xadd
2404   {MT@##1@csname MT@##1@context\endcsname font@list}%
2405   {\font@name,}%
2406 }%
2407 }

```

`\MT@register@font@cx` For each feature, add the current font to the list, unless we didn't set it up.

```

2408 \def\MT@register@font@cx{%
2409   \MT@map@clist@c\MT@active@features{%
2410     \MT@exp@cs\ifx{MT@\@nameuse{MT@abbr@##1}}\relax\else
2411       \MT@exp@cs\MT@xadd
2412       {MT@##1@csname MT@##1@context\endcsname font@list}%
2413       {\MT@font,}%
2414     \def\@tempa{##1}%
2415     \MT@exp@cs\MT@map@tlist@c{MT@##1@doc@contexts}\MT@maybe@rem@from@list
2416   \fi
2417 }%
2418 }

```

`\MT@maybe@rem@from@list` Recurse through all context font lists of the document and remove the font, unless it's the current context.

```

2419 \def\MT@maybe@rem@from@list#1{%
2420   \MT@ifstreql{\@tempa/#1}{\@tempa/\csname MT@\@tempa @context\endcsname}\relax{%
2421     \expandafter\MT@exp@one@n\expandafter\MT@rem@from@clist\expandafter
2422     \MT@font \csname MT@\@tempa @#1font@list\endcsname
2423   }%
2424 }

```

`\microtypecontext` The user may change the context, so that different setups are possible. This is especially useful for multi-lingual documents.

Inside the preamble, it shouldn't actually do anything but remember it for later.

```

2425 \def\microtypecontext#1{\MT@addto@setup{\microtypecontext{#1}}}
2426 \MT@addto@setup{%
2427   \DeclareRobustCommand\microtypecontext[1]{%
2428     \MT@setup@contexts
2429     \let\MT@reset@context\relax
2430     \setkeys{MTC}{#1}%
2431     \selectfont
2432     \MT@reset@context
2433   }%
2434 }

```

`\textmicrotypecontext` This is just a wrapper around `\microtypecontext`.

```

2435 \DeclareRobustCommand\textmicrotypecontext[2]{\microtypecontext{#1}#2}

```

`\MT@reset@context` We have to reset the font at the end of the group, provided there actually was a change.

`\MT@reset@context@`

```

2436 \def\MT@reset@context@{%
2437   \MT@vinfo{<<< Resetting contexts\on@line
2438   <debug> \MessageBreak= \MT@pr@context/\MT@ex@context
2439   <debug>           /\MT@tr@context/\MT@kn@context/\MT@sp@context
2440   }%
2441   \selectfont
2442 }

```

`\MT@setup@contexts` The first time `\microtypecontext` is called, we initialise the context lists and redefine the commands used in `\pickup@font`.

```

2443 \def\MT@setup@contexts{%
2444   \MT@map@clist@c\MT@active@features
2445   {\MT@gl@et@nc{MT@##1@font@list}\MT@font@list}%

```

```

2446 \MT@gllet\MT@check@font\MT@check@font@cx
2447 \MT@gllet\MT@register@font\MT@register@font@cx
2448 \MT@gllet\MT@register@subst@font\MT@register@subst@font@cx
2449 \MT@gllet\MT@setup@contexts\relax
2450 }

```

Define context keys.

```

2451 \MT@map@clist@c\MT@features@long{%
2452 \define@key{MTC}{#1}[]{}%
2453 \edef\@tempb{\@nameuse{MT@rbba#1}}%
2454 \MT@exp@one@n\MT@in@clist\@tempb\MT@active@features
2455 \ifMT@inlist@

```

Using an empty context is only asking for trouble, therefore we choose the ‘@’ instead (hoping for the L^AT_EX users’ natural awe of this character).

```

2456 \MT@ifempty{#1}{\def\MT@val{0}}{\def\MT@val{#1}}%
2457 \MT@exp@cs\ifx{MT@\@tempb @context}\MT@val
2458 (debug) \MT@info{1}{>>> no change of #1 context: \MT@val}%
2459 \else
2460 \MT@vinfo{>>> Changing #1 context to \MT@val'\MessageBreak\on@line
2461 (debug) \space(previous: \@nameuse{MT@\@tempb @context}')%
2462 }%
2463 \def\MT@reset@context{\aftergroup\MT@reset@context@}%

```

The next time we see the font, we have to reset *all* factors.

```

2464 \MT@gllet@nn{MT@reset@\@tempb @codes}{MT@reset@\@tempb @codes@}%

```

We must also keep track of all contexts in the document.

```

2465 \expandafter\MT@exp@one@n\expandafter\MT@in@tlist\expandafter
2466 \MT@val \curname MT@\@tempb @doc@contexts\endcurname
2467 \ifMT@inlist@ \else
2468 \MT@exp@cs\MT@xadd{MT@\@tempb @doc@contexts}{\MT@val}}%
2469 (debug) \MT@info{1}{||| added #1 context: \@nameuse{MT@\@tempb @doc@contexts}}%
2470 \fi
2471 \MT@edef@n{MT@\@tempb @context}{\MT@val}%
2472 \fi
2473 \fi
2474 }%
2475 }

```

`\MT@pr@context` Initialise the contexts.

`\MT@ex@context` 2476 `\MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{}`%

`\MT@tr@context` 2477 `\MT@def@n{MT@#1@context}{@}%`

`\MT@sp@context` 2478 `\MT@def@n{MT@#1@doc@contexts}{@}%`

2479 }

`\MT@kn@context` 2480 `\let\MT@extra@context\empty`

`\MT@pr@doc@contexts`

`\MT@ex@doc@contexts`

`\MT@tr@doc@contexts`

`\MT@sp@doc@contexts`

`\MT@kn@doc@contexts`

`\DeclareMicrotypeSet`

`\MT@extra@context`

`\DeclareMicrotypeSet*`

14.3 Configuration

14.3.1 Font sets

Calling this macro will create a comma list for every font attribute of the form: `\MT{feature}list@{attribute}@{set name}`. If the optional argument is empty, lists for all available features will be created.

The third argument must be a list of key=value pairs. If a font attribute is not specified, we define the corresponding list to `\relax`, so that it does not constitute a constraint.

```

2481 \def\DeclareMicrotypeSet{%
2482 \ifstar

```

```

2483 \MT@DeclareSetAndUseIt
2484 \MT@DeclareSet
2485 }

\MT@DeclareSet
2486 \newcommand\MT@DeclareSet[3] [] {%
2487 \KV@sp@def\@tempa{#1}%
2488 \MT@ifempty\@tempa{%
2489 \MT@map@clist@c\MT@features{\MT@declare@sets{##1}{#2}{#3}}}%
2490 }{%
2491 \MT@map@clist@c\@tempa{%
2492 \KV@sp@def\@tempa{##1}%
2493 \MT@ifempty\@tempa\relax{%
2494 \MT@ifis@feature{set declaration `#2'}{%
2495 \MT@exp@one@n\MT@declare@sets
2496 {\csname MT@rbba@\@tempa\endcsname}{#2}{#3}%
2497 }%
2498 }%
2499 }}%
2500 }%
2501 }

\MT@DeclareSetAndUseIt
2502 \newcommand\MT@DeclareSetAndUseIt[3] [] {%
2503 \MT@DeclareSet[#1]{#2}{#3}%
2504 \UseMicrotypeSet[#1]{#2}%
2505 }

\MT@curr@set@name We need to remember the name of the set currently being declared.
2506 \let\MT@curr@set@name\empty

\MT@declare@sets Define the current set name and parse the keys.
2507 \def\MT@declare@sets#1#2#3{%
2508 \KV@sp@def\MT@curr@set@name{#2}%
2509 \MT@ifdefined@n@T{MT@#1@set@@\MT@curr@set@name}{%
2510 \MT@warning{Redefining \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2511 \MT@glet@nc{MT@#1list@size@\MT@curr@set@name}\empty
2512 }%
2513 \MT@glet@nc{MT@#1@set@@\MT@curr@set@name}\empty
2514 (debug)\MT@edinfo{1}{declaring \@nameuse{MT@abbr@#1} set `~\MT@curr@set@name'}%
2515 \setkeys{MT@#1@set}{#3}%
2516 }

\MT@define@set@key@ <#1> = font axis, <#2> = feature.
2517 \def\MT@define@set@key@#1#2{%
2518 \define@key{MT@#2@set}{#1} [] {%
2519 \MT@glet@nc{MT@#2list@#1@\MT@curr@set@name}\empty
2520 \MT@map@clist@n{##1}{%
2521 \KV@sp@def\MT@val{###1}%
2522 \MT@get@highlevel{#1}%

We do not add the expanded value to the list ...
2523 \MT@exp@two@n\g@addto@macro
2524 {\csname MT@#2list@#1@\MT@curr@set@name\expandafter\endcsname}%
2525 {\MT@val},}%
2526 }%

... but keep in mind that the list has to be expanded at the end of the preamble.
2527 \expandafter\g@addto@macro\expandafter\MT@font@sets
2528 \csname MT@#2list@#1@\MT@curr@set@name\endcsname
2529 (debug)\MT@edinfo@n1{1}{-- #1: \@nameuse{MT@#2list@#1@\MT@curr@set@name}}%
2530 }%

```

2531 }

\MT@get@highlevel Saying, for instance, ‘family=rm*’ or ‘shape=bf*’ will expand to \rmdefault resp. \bfdefault.

2532 \def\MT@get@highlevel#1{%

2533 \expandafter\MT@test@ast\MT@val*\@nil\relax{%

And ‘family = *’ will become \familydefault.

2534 \MT@ifempty\@tempa{\def\@tempa{#1}}\relax

2535 \edef\MT@val{\expandafter\noexpand\csname \@tempa default\endcsname}%

In contrast to earlier version, these values will not be expanded immediately but at the end of the preamble.

2536 }%

2537 }

\MT@test@ast It the last character is an asterisk, execute the second argument, otherwise the first one.

2538 \def\MT@test@ast#1*#2\@nil{%

2539 \def\@tempa{#1}%

2540 \MT@ifempty{#2}%

2541 }

\MT@font@sets Fully expand the font specification and fix catcodes for all font sets.

\MT@fix@font@set 2542 \let\MT@font@sets\empty

2543 \def\MT@fix@font@set#1{%

2544 \xdef#1{#1}%

2545 \global\@onelevel@sanitize#1%

2546 }

\MT@define@set@key@size size requires special treatment.

2547 \def\MT@define@set@key@size#1{%

2548 \define@key{MT@#1@set}{size}[]{%

2549 \MT@map@clist@n{##1}{%

2550 \KV@sp@def\MT@val{###1}%

2551 \expandafter\MT@get@range\MT@val--\@nil

2552 \ifx\MT@val\relax \else

2553 \MT@exp@cs\MT@xadd

2554 {MT@#1list@size@\MT@curr@set@name}%

2555 {{{\MT@lower}{\MT@upper}\relax}}%

2556 \fi

2557 }%

2558 *<debug>*\MT@info@n1{1}{-- size: \@nameuse{MT@#1list@size@\MT@curr@set@name}}%

2559 }%

2560 }

Font sizes may also be specified as ranges. This has been requested by Andreas Böhmann, who has also offered valuable help in implementing this. Now, it is for instance possible to set up different lists for fonts with optical sizes. (The MinionPro project is trying to do this for the OpenType version of Adobe’s Minion. See <http://developer.berlios.de/projects/minionpro/>.)

\MT@get@range Ranges will be stored as triplets of {<lower bound>}{<upper bound>}{<list name>}.

\MT@upper For simple sizes, the upper boundary is −1.

\MT@lower 2561 \def\MT@get@range#1-#2-#3\@nil{%

2562 \MT@ifempty{#1}{%

2563 \MT@ifempty{#2}{%

2564 \let\MT@val\relax

2565 }%

2566 \def\MT@lower{0}%

```

2567     \def\MT@val{#2}%
2568     \MT@get@size
2569     \edef\MT@upper{\MT@val}%
2570     }%
2571   }{%
2572     \def\MT@val{#1}%
2573     \MT@get@size
2574     \ifx\MT@val\relax \else
2575       \edef\MT@lower{\MT@val}%
2576       \MT@ifempty{#2}{%
2577         \MT@ifempty{#3}%
2578       }{\def\MT@upper{-1}}%

```

2048 pt is \TeX 's maximum font size.

```

2579     {\def\MT@upper{2048}}%
2580   }{%
2581     \def\MT@val{#2}%
2582     \MT@get@size
2583     \ifx\MT@val\relax \else
2584       \MT@ifdim\MT@lower>\MT@val{%
2585         \MT@error{%
2586           Invalid size range (\MT@lower\space > \MT@val) in font set
2587           '~\MT@curr@set@name'.\MessageBreak Swapping sizes}}%
2588       \edef\MT@upper{\MT@lower}%
2589       \edef\MT@lower{\MT@val}%
2590     }{%
2591       \edef\MT@upper{\MT@val}%
2592     }%
2593     \MT@ifdim\MT@lower=\MT@upper
2594     {\def\MT@upper{-1}}%
2595     \relax
2596   \fi
2597 }%
2598 \fi
2599 }%
2600 }

```

`\MT@get@size` Translate a size selection command and normalise it.

```

2601 \def\MT@get@size{%
    A single star would mean \sizedefault, which doesn't exist, so we define it to be
    \normalsize.
2602   \if*\MT@val\relax
2603     \def\@tempa{\normalsize}%
2604   \else
2605     \MT@let@cn\@tempa{\MT@val}%
2606   \fi
2607   \ifx\@tempa\relax \else

```

The `resize` solution of parsing `\@setfontsize` does not work with the AMS classes, among others. I hope my hijacking doesn't do any harm. We redefine `\set@fontsize`, and not `\@setfontsize` because some classes might define the size selection commands by simply using `\fontsize` (e. g., the `a0poster` class).

```

2608   \begingroup
2609     \def\set@fontsize##1##2##3##4\@nil{\endgroup\def\MT@val{##2}}%
2610     \@tempa\@nil
2611   \fi

```

Test whether we finally got a number or dimension so that we can strip the 'pt' (`\@defaultunits` and `\strip@pt` are kernel macros).

```

2612   \MT@ifdimen\MT@val{%

```



```

2613 \defaultunits\@tempdima\MT@val pt\relax\@nnil
2614 \edef\MT@val{\strip@pt\@tempdima}%
2615 }{%
2616 \MT@warning{Could not parse font size `\'MT@val'\MessageBreak
2617 in font set `\'MT@curr@set@name'}%
2618 \let\MT@val\relax
2619 }%
2620 }

```

\MT@define@set@key@font

```

2621 \def\MT@define@set@key@font#1{%
2622 \define@key{MT@#1@set}{font}[]{%
2623 \MT@glet@nc{MT@#1list@font@\'MT@curr@set@name}\@empty
2624 \MT@map@clist@n{##1}{%
2625 \KV@esp@def\MT@val{###1}%
2626 \MT@ifstreq\MT@val{*/*/*/*/*}\relax
2627 \expandafter\MT@get@font\MT@val////\@nil
2628 \MT@exp@two@n@g@addto@macro
2629 {\csname MT@#1list@font@\'MT@curr@set@name\expandafter\endcsname}%
2630 {\MT@val,}%
2631 }%
2632 \expandafter\g@addto@macro\expandafter\MT@font@sets
2633 \csname MT@#1list@font@\'MT@curr@set@name\endcsname
2634 <debug>\MT@info@n1{1}{-- font: \nameuse{MT@#1list@font@\'MT@curr@set@name}}%
2635 }%
2636 }

```

\MT@get@font Translate any asterisks.

```

2637 \def\MT@get@font#1/#2/#3/#4/#5/#6\@nil{%
2638 \MT@get@font@{#1}{#2}{#3}{#4}{#5}{0}%
2639 \ifx\MT@val\relax\def\MT@val{0}\fi
2640 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val}%
2641 \let\MT@val\@tempb
2642 }

```

\MT@get@font@ Helper macro, also used by \MT@get@font@and@size.

```

2643 \def\MT@get@font@#1#2#3#4#5#6{%
2644 \let\@tempb\@empty
2645 \def\MT@temp{#1/#2/#3/#4/#5}%
2646 \MT@get@axis{encoding}{#1}%
2647 \MT@get@axis{family}{#2}%
2648 \MT@get@axis{series}{#3}%
2649 \MT@get@axis{shape}{#4}%
2650 \ifnum#6>\z@\edef\@tempb{\@tempb*}\fi
2651 \MT@ifempty{#5}{%
2652 \MT@warn@axis@empty{size}{\string\normalsize}%
2653 \def\MT@val{*}%
2654 }{%
2655 \def\MT@val{#5}%
2656 }%
2657 \MT@get@size
2658 }

```

\MT@get@axis

```

2659 \def\MT@get@axis#1#2{%
2660 \def\MT@val{#2}%
2661 \MT@get@highlevel{#1}%
2662 \MT@ifempty\MT@val{%
2663 \MT@warn@axis@empty{#1}{\csname #1default\endcsname}%
2664 \expandafter\def\expandafter\MT@val\expandafter{\csname #1default\endcsname}%
2665 }\relax
2666 \expandafter\g@addto@macro\expandafter\@tempb\expandafter{\MT@val/}%

```

2667 }

\MT@warn@axis@empty

```
2668 \def\MT@warn@axis@empty#1#2{%
2669   \MT@warning{#1 axis is empty in font specification\MessageBreak
2670   ~\MT@temp'. Using ~#2' instead}%
2671 }
```

We can finally assemble all pieces to define \DeclareMicrotypeSet's keys. They are also used for \DisableLigatures.

```
2672 \MT@exp@one@n\MT@map@clist@n{\MT@features,nl}{%
2673   \MT@define@set@key@{encoding}{#1}%
2674   \MT@define@set@key@{family}   {#1}%
2675   \MT@define@set@key@{series}   {#1}%
2676   \MT@define@set@key@{shape}    {#1}%
2677   \MT@define@set@key@size       {#1}%
2678   \MT@define@set@key@font       {#1}%
2679 }
```

\UseMicrotypeSet

To use a particular set we simply redefine MT@(*feature*)@setname. If the optional argument is empty, set names for all features will be redefined.

```
2680 \renewcommand*\UseMicrotypeSet[2][]{%
2681   \KV@sp@def\@tempa{#1}%
2682   \MT@ifempty\@tempa{%
2683     \MT@map@clist@c\MT@features{{\MT@use@set{##1}{#2}}}%
2684   }%
2685   \MT@map@clist@c\@tempa{%
2686     \KV@sp@def\@tempa{##1}%
2687     \MT@ifempty\@tempa\relax{%
2688       \MT@is@feature{activation of set ~#2'}{%
2689         \MT@exp@one@n\MT@use@set
2690         {~\csname MT@rbba@\@tempa\endcsname}{#2}%
2691       }%
2692     }%
2693   }}%
2694 }%
2695 }
```

\MT@pr@setname Only use sets that have been declared.

```
\MT@ex@setname 2696 \def\MT@use@set#1#2{%
\MT@tr@setname 2697   \KV@sp@def\@tempa{#2}%
2698   \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@sp@setname 2699   \MT@xdef@n{MT@#1@setname}{\@tempa}%
\MT@kn@setname 2700 }%
\MT@use@set 2701   \MT@ifdefined@n@TF{MT@#1@setname}\relax{%
2702     \MT@xdef@n{MT@#1@setname}{\@nameuse{MT@default@#1@set}}%
2703   }%
2704   \MT@error{%
2705     The \@nameuse{MT@abbr@#1} set ~\@tempa' is undeclared.\MessageBreak
2706     Using set ~\@nameuse{MT@#1@setname}' instead}{}%
2707 }%
2708 }
```

\DeclareMicrotypeSetDefault

This command can be used in the main configuration file to declare the default font set, in case no set is specified in the package options.

```
2709 \renewcommand*\DeclareMicrotypeSetDefault[2][]{%
2710   \KV@sp@def\@tempa{#1}%
2711   \MT@ifempty\@tempa{%
2712     \MT@map@clist@c\MT@features{{\MT@set@default@set{##1}{#2}}}%
2713   }%
2714   \MT@map@clist@c\@tempa{%
```

```

2715 \KV@sp@def\@tempa{##1}%
2716 \MT@ifempty\@tempa\relax{%
2717 \MT@is@feature{declaration of default set `#2'}{%
2718 \MT@exp@one@n\MT@set@default@set
2719 {\csname MT@rbba@\@tempa\endcsname}{#2}%
2720 }%
2721 }%
2722 }}%
2723 }%
2724 }

\MT@default@pr@set
\MT@default@ex@set 2725 \def\MT@set@default@set#1#2{%
\MT@default@tr@set 2726 \KV@sp@def\@tempa{#2}%
2727 \MT@ifdefined@n@TF{MT@#1@set@@\@tempa}{%
\MT@default@sp@set 2728 debug\MT@edinfo{1}{declaring default \@nameuse{MT@abbr@#1} set \@tempa'}%
\MT@default@kn@set 2729 \MT@xdef@n{MT@default@#1@set}{\@tempa}%
\MT@set@default@set 2730 }%
2731 \MT@error{%
2732 The \@nameuse{MT@abbr@#1} set \@tempa' is not declared.\MessageBreak
2733 Cannot make it the default set. Using set\MessageBreak `all' instead{}}%
2734 \MT@xdef@n{MT@default@#1@set}{all}%
2735 }%
2736 }

```

14.3.2 Variants and aliases

`\DeclareMicrotypeVariants` Specify suffixes for variants (see `fontname/variants.map`). The starred version appends to the list.

```

\MT@variants
2737 \let\MT@variants\@empty
2738 \def\DeclareMicrotypeVariants{%
2739 \ifstar
2740 \MT@DeclareVariants
2741 {\let\MT@variants\@empty\MT@DeclareVariants}%
2742 }

```

`\MT@DeclareVariants`

```

2743 \def\MT@DeclareVariants#1{%
2744 \MT@map@clist@n{#1}{%
2745 \KV@sp@def\@tempa{##1}%
2746 \@onelevel@sanitize\@tempa
2747 \xdef\MT@variants{\MT@variants{\@tempa}}%
2748 }%
2749 }

```

`\DeclareMicrotypeAlias` This can be used to set an alias name for a font, so that the file and the settings for the aliased font will be loaded.

```

2750 \renewcommand*\DeclareMicrotypeAlias[2]{%
2751 \KV@sp@def\@tempa{#1}%
2752 \KV@sp@def\@tempb{#2}%
2753 \@onelevel@sanitize\@tempb
2754 \MT@ifdefined@n@T{MT@\@tempa @alias}{%
2755 \MT@warning{Alias font family \@tempb' will override
2756 alias \@nameuse{MT@\@tempa @alias}'\MessageBreak
2757 for font family \@tempa'}}%
2758 \MT@xdef@n{MT@\@tempa @alias}{\@tempb}%

```

If we encounter this command while a font is being set up, we also set the alias for the current font so that if `\DeclareMicrotypeAlias` has been issued inside a configuration file, the configuration file for the alias font will be loaded, too.

```

2759 \MT@ifdefined@c@T\MT@family{%
2760 (debug)\MT@edinfo{1}{Activating alias font `\'@tempb' for `\'MT@family'}%
2761 \MT@glet\MT@familyalias\@tempb
2762 }%
2763 }

```

`\LoadMicrotypeFile` May be used to load a configuration file manually.

```

2764 \def\LoadMicrotypeFile#1{%
2765 \KV@sp@def\@tempa{#1}%
2766 \onelevel@sanitize\@tempa
2767 \MT@exp@one@n\MT@in@clist\@tempa\MT@file@list
2768 \ifMT@inlist@
2769 \MT@vinfo{... Configuration file mt-\@tempa.cfg already loaded}%
2770 \else
2771 \MT@xadd\MT@file@list{\@tempa,}%
2772 \MT@begin@catcodes
2773 \InputIfFileExists{mt-\@tempa.cfg}{%
2774 \edef\MT@curr@file{mt-\@tempa.cfg}%
2775 \MT@vinfo{... Loading configuration file \MT@curr@file}%
2776 }{%
2777 \MT@warning{... Configuration file mt-\@tempa.cfg\MessageBreak
2778 does not exist}%
2779 }%
2780 \MT@end@catcodes
2781 \fi
2782 }

```

14.3.3 Disabling ligatures

`\DisableLigatures` This is really simple now: we can re-use the set definitions of `\DeclareMicrotypeSet`; there can only be one set, which we'll call 'no ligatures'.

`\MT@n1@setname` The optional argument may be used to disable selected ligatures only.

```

\MT@n1@ligatures 2783 \MT@requires@pdftex5{
2784 \def\DisableLigatures{%
2785 \MT@begin@catcodes
2786 \MT@DisableLigatures
2787 }
2788 \newcommand*\MT@DisableLigatures[2][]{%
2789 \MT@ifempty{#1}\relax{\gdef\MT@n1@ligatures{#1}}%
2790 \xdef\MT@active@features{\MT@active@features,nl}%
2791 \global\MT@noligaturestrue
2792 \MT@declare@sets{nl}{no ligatures}{#2}%
2793 \gdef\MT@n1@setname{no ligatures}%
2794 \MT@end@catcodes
2795 }
2796 }{

```

If pdf_{TEX} is too old, we throw an error.

```

2797 \renewcommand*\DisableLigatures[2][]{%
2798 \MT@error{Disabling ligatures of a font is only possible\MessageBreak
2799 with pdftex version 1.30 or newer.\MessageBreak
2800 Ignoring \string\DisableLigatures}{Upgrade pdftex.}%
2801 }
2802 }

```

14.3.4 Interaction with babel

`\DeclareMicrotypeBabelHook` Declare the context that should be loaded when a babel language is selected. The command will not check whether a previous declaration will be overwritten.

```

2803 \def\DeclareMicrotypeBabelHook#1#2{%
2804   \MT@map@clist@n{#1}{%
2805     \KV@esp@def\@tempa{##1}%
2806     \MT@gdef@n{MT@babel@{\@tempa}{#2}%
2807   }%
2808 }

```

14.3.5 Fine tuning

The commands `\SetExpansion` and `\SetProtrusion` provide an interface for setting the character protrusion resp. expansion factors for a set of fonts.

`\SetProtrusion` This macro accepts three arguments: [options,] set of font attributes and list of character protrusion factors.

A new macro called `\MT@pr@c@{name}` will be defined to be `<#3>` (i. e., the list of characters, not expanded).

```

2809 \def\SetProtrusion{%
2810   \MT@begin@catcodes
2811   \MT@SetProtrusion
2812 }

```

`\MT@SetProtrusion` We want the catcodes to be correct even if this is called in the preamble.

```

\MT@pr@c@name 2813 \newcommand*\MT@SetProtrusion[3] [] {%

```

`\MT@extra@context` 2814 `\let\MT@extra@context\@empty`

`\MT@permutelist` Parse the optional first argument. We first have to know the name before we can deal with the extra options.

```

2815   \MT@set@named@keys{MT@pr@c}{#1}%
2816   <debug> \MT@info{1}{creating protrusion list `~\MT@pr@c@name'}%
2817   \def\MT@permutelist{pr@c}%
2818   \setkeys{MT@cfg}{#2}%

```

We have parsed the second argument, and can now define macros for all permutations of the font attributes to point to `\MT@pr@c@{name}`, ...

```

2819   \MT@permute

```

... which we can now define to be `<#3>`. Here, as elsewhere, we have to make the definitions global, since they will occur inside a group.

```

2820   \MT@gdef@n{MT@pr@c@\MT@pr@c@name}{#3}%
2821   \MT@end@catcodes
2822 }

```

`\SetExpansion` `\SetExpansion` only differs in that it allows some extra options (stretch, shrink, step, auto).

```

2823 \def\SetExpansion{%
2824   \MT@begin@catcodes
2825   \MT@SetExpansion
2826 }

```

`\MT@SetExpansion`

```

\MT@ex@c@name 2827 \newcommand*\MT@SetExpansion[3] [] {%

```

`\MT@extra@context` 2828 `\let\MT@extra@context\@empty`

`\MT@permutelist` 2829 `\MT@set@named@keys{MT@ex@c}{#1}%`

```

2830 \MT@ifdefined@n@T{MT@ex@c@\MT@ex@c@name @factor}{%

```

```

2831   \ifnum\csname MT@ex@c@\MT@ex@c@name @factor\endcsname > \@m

```

```

2832     \MT@warning@n1{Expansion factor \number\@nameuse{MT@ex@c@\MT@ex@c@name @factor}

```

```

2833       too large in list\MessageBreak `~\MT@ex@c@name'. Setting it to the

```

```

2834       maximum of 1000}%

```

```

2835   \MT@glet@nc{MT@ex@c@\MT@ex@c@name @factor}\@m

```

```

2836     \fi
2837   }%
2838   (debug)\MT@dinfo{1}{creating expansion list `~\MT@ex@c@name'}%
2839   \def\MT@permutelist{ex@c}%
2840   \setkeys{MT@cfg}{#2}%
2841   \MT@permute
2842   \MT@gdef@n{MT@ex@c@~\MT@ex@c@name}{#3}%
2843   \MT@end@catcodes
2844 }

\SetTracking
2845 \def\SetTracking{%
2846   \MT@begin@catcodes
2847   \MT@SetTracking
2848 }

\MT@SetTracking    Third argument may be empty.
2849 \newcommand*\MT@SetTracking[3][]{%
2850   \let\MT@extra@context\@empty
2851   \MT@set@named@keys{MT@tr@c}{#1}%
2852   (debug)\MT@dinfo{1}{creating tracking list `~\MT@tr@c@name'}%
2853   \def\MT@permutelist{tr@c}%
2854   \setkeys{MT@cfg}{#2}%
2855   \MT@permute
2856   \KV@sp@def\@tempa{#3}%
2857   \MT@ifempty\@tempa\relax{%
2858     \MT@ifint\@tempa
2859     {\MT@gdef@n{MT@tr@c@~\MT@tr@c@name}{\@tempa}}%
2860     {\MT@warning{Value `~\@tempa' is not a number in\MessageBreak
2861       tracking set `~\MT@curr@set@name'}}}%
2862   \MT@end@catcodes
2863 }

\SetExtraSpacing
2864 \def\SetExtraSpacing{%
2865   \MT@begin@catcodes
2866   \MT@SetExtraSpacing
2867 }

\MT@SetExtraSpacing
\MT@sp@c@name 2868 \newcommand*\MT@SetExtraSpacing[3][]{%
\MT@extra@context 2869   \let\MT@extra@context\@empty
2870   \MT@set@named@keys{MT@sp@c}{#1}%
\MT@permutelist 2871   (debug)\MT@dinfo{1}{creating spacing list `~\MT@sp@c@name'}%
2872   \def\MT@permutelist{sp@c}%
2873   \setkeys{MT@cfg}{#2}%
2874   \MT@permute
2875   \MT@gdef@n{MT@sp@c@~\MT@sp@c@name}{#3}%
2876   \MT@end@catcodes
2877 }

\SetExtraKerning
2878 \def\SetExtraKerning{%
2879   \MT@begin@catcodes
2880   \MT@SetExtraKerning
2881 }

\MT@SetExtraKerning
\MT@kn@c@name 2882 \newcommand*\MT@SetExtraKerning[3][]{%
2883   \let\MT@extra@context\@empty
\MT@extra@context 2884   \MT@set@named@keys{MT@kn@c}{#1}%
\MT@permutelist 2885   (debug)\MT@dinfo{1}{creating kerning list `~\MT@kn@c@name'}%

```

```

2886 \def\MT@permutelist{kn@c}%
2887 \setkeys{MT@cfg}{#2}%
2888 \MT@permute
2889 \MT@gdef@n{MT@kn@c@MT@kn@c@name}{#3}%
2890 \MT@end@catcodes
2891 }

```

\MT@set@named@keys We first set the name (if specified), then remove it from the list, and set the remaining keys.

\MT@options

```

2892 \def\MT@set@named@keys#1#2{%
2893 \def\x##1name=##2,##3\@nil{%
2894 \setkeys{#1}{name=##2}%
2895 \gdef\MT@options{##1##3}%
2896 \MT@rem@from@clist{name=}\MT@options
2897 }%
2898 \x#2,name=,\@nil
2899 \@expandtwoargs\setkeys{#1}\MT@options
2900 }

```

\MT@define@code@key Define the keys for the configuration lists (which are setting the codes, in pdfTeX speak).

```

2901 \def\MT@define@code@key#1#2{%
2902 \define@key{MT@#2}{#1}[]{%
2903 \@tempcnta=\@ne
2904 \MT@map@clist@n{##1}{%
2905 \KV@sp@def\MT@val{###1}%

```

Here, too, we allow for something like ‘bf*’. It will be expanded immediately.

```

2906 \MT@get@highlevel{#1}%
2907 \MT@edef@n{MT@temp#1\the\@tempcnta}{\MT@val}%
2908 \advance\@tempcnta \@ne
2909 }%
2910 }%
2911 }

```

\MT@define@code@key@size \MT@tempsize must be in a \csname, so that it is at least \relax, not undefined.

```

2912 \def\MT@define@code@key@size#1{%
2913 \define@key{MT@#1}{size}[]{%
2914 \MT@map@clist@n{##1}{%
2915 \KV@sp@def\MT@val{###1}%
2916 \expandafter\MT@get@range\MT@val--\@nil
2917 \ifx\MT@val\relax \else
2918 \MT@exp@cs\MT@xadd{MT@tempsize}%
2919 {{{\MT@lower}{\MT@upper}{\MT@curr@set@name}}}%
2920 \fi
2921 }%
2922 }%
2923 }

```

\MT@define@code@key@font

```

2924 \def\MT@define@code@key@font#1{%
2925 \define@key{MT@#1}{font}[]{%
2926 \MT@map@clist@n{##1}{%
2927 \KV@sp@def\MT@val{###1}%
2928 \MT@ifstreq\MT@val*{\def\MT@val{*/**/*/*}}\relax
2929 \expandafter\MT@get@font@and@size\MT@val///// \@nil
2930 \MT@xdef@n{MT@MT@permutelist @\@tempb\MT@extra@context}%
2931 {\csname MT@MT@permutelist @name\endcsname}%
2932 <debug>\MT@edinfo@n1{1}{initialising: use list for font \@tempb=\MT@val
2933 <debug> \ifx\MT@extra@context\@empty\else\MessageBreak
2934 <debug> (context: \MT@extra@context)\fi}%
2935 \MT@exp@cs\MT@xaddb

```

```

2936      {MT@MT@permutelist @\@tempb\MT@extra@context @sizes}%
2937      {{{\MT@eval}{\m@ne}{\MT@curr@set@name}}}%
2938    }%
2939  }%
2940 }

```

`\MT@get@font@and@size` Translate any asterisks and split off the size.

```

2941 \def\MT@get@font@and@size#1/#2/#3/#4/#5/#6\@nil{%
2942   \MT@get@font@{#1}{#2}{#3}{#4}{#5}{1}%
2943 }
2944 \MT@define@code@key{encoding}{cfg}
2945 \MT@define@code@key{family} {cfg}
2946 \MT@define@code@key{series} {cfg}
2947 \MT@define@code@key{shape} {cfg}
2948 \MT@define@code@key@size {cfg}
2949 \MT@define@code@key@font {cfg}

```

`\MT@define@opt@key`

```

2950 \def\MT@define@opt@key#1#2{%
2951   \define@key{MT@#1@c}{#2}[]{\MT@ifempty{##1}\relax{%
2952     \MT@xdef@n{MT@#1@c@MT@curr@set@name @#2}{##1}}}%
2953 }

```

The options in the optional first argument.

```

2954 \MT@map@clist@c\MT@features{%

```

Use file name and line number as the list name if the user didn't bother to invent one.

```

2955   \define@key{MT@#1@c}{name}[]{%
2956     \MT@ifempty{##1}{%
2957       \MT@edef@n{MT@#1@c@name}{\MT@curr@file/\the\inputlineno}%
2958     }{%
2959       \MT@edef@n{MT@#1@c@name}{##1}%
2960       \MT@ifdefined@n@T{MT@#1@c@csname MT@#1@c@name\endcsname}{%
2961         \MT@warning{Redefining \@nameuse{MT@abbr@#1} list ~\@nameuse{MT@#1@c@name}'}%
2962       }%
2963     }%
2964     \MT@let@cn\MT@curr@set@name{MT@#1@c@name}%
2965   }%
2966   \MT@define@opt@key{#1}{load}%
2967   \MT@define@opt@key{#1}{factor}%
2968   \MT@define@opt@key{#1}{preset}%
2969   \MT@define@opt@key{#1}{inputenc}%

```

Only one context is allowed. This might change in the future.

```

2970   \define@key{MT@#1@c}{context}[]{\MT@ifempty{##1}\relax{\def\MT@extra@context{##1}}}%
2971 }

```

Automatically enable font copying if we find a protrusion or expansion context. After the preamble, check whether font copying is enabled. For older pdfTeX versions, disallow. Also disable for luaTeX.

```

2972 \MT@requires@pdftex7{
2973   (*lua)
2974   \MT@requires@luatex{
2975     \define@key{MT@ex@c}{context}[]{%
2976       \MT@error{Expansion contexts currently don't work with luatex.\MessageBreak
2977         Ignoring `context' key\on@line}%
2978       {Use pdftex instead.}%
2979     }
2980   }{
2981     (/lua)

```



```

2982 \define@key{MT@ex@c}{context}[]{%
2983 \MT@ifempty{#1}\relax{%
2984 \MT@gllet\MT@copy@font\MT@copy@font@
2985 \def\MT@extra@context{#1}%
2986 }%
2987 }
2988 \MT@addto@setup{%
2989 \define@key{MT@ex@c}{context}[]{%
2990 \ifx\MT@copy@font\MT@copy@font@
2991 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
2992 \else
2993 \MT@error{\MT@MT\space isn't set up for expansion contexts.\MessageBreak
2994 Ignoring `context' key\on@line}%
2995 {Either move the settings inside the preamble,\MessageBreak
2996 or load the package with the `copyfonts' option.}%
2997 \fi
2998 }
2999 }

```

Protrusion contexts *may* also work without copying the font, so we don't issue an error but only a warning. The problem is that pdfTeX only allows one set of protrusion factors for a given font within one paragraph (those that are in effect at the end of the paragraph will be in effect for the whole paragraph). When different fonts are loaded – like in the example with the footnote markers – we don't need to copy the fonts.

```

3000 \define@key{MT@pr@c}{context}[]{%
3001 \MT@ifempty{#1}\relax{%
3002 \MT@gllet\MT@copy@font\MT@copy@font@
3003 \def\MT@extra@context{#1}%
3004 }%
3005 }
3006 \MT@addto@setup{%
3007 \define@key{MT@pr@c}{context}[]{%
3008 \MT@ifempty{#1}\relax{\def\MT@extra@context{#1}}%
3009 \ifx\MT@copy@font\MT@copy@font@\else
3010 \MT@warning@n1{If protrusion contexts don't work as expected,
3011 \MessageBreak load the package with the `copyfonts' option}%
3012 \fi
3013 }
3014 }
3015 \lua }
3016 }{
3017 \define@key{MT@ex@c}{context}[]{%
3018 \MT@error{Expansion contexts only work with pdftex 1.40.4\MessageBreak
3019 or later. Ignoring `context' key\on@line}%
3020 {Upgrade pdftex.}%
3021 }
3022 }

```

\MT@warn@nodim

```

3023 \def\MT@warn@nodim#1{%
3024 \MT@warning{\`@tempa' is not a dimension.\MessageBreak
3025 Ignoring it and setting values relative to\MessageBreak #1}%
3026 }

```

Protrusion codes may be relative to character width, or to any dimension.

```

3027 \define@key{MT@pr@c}{unit}[character]{%
3028 \MT@gllet@nc{MT@pr@c@\MT@curr@set@name @unit}\@empty
3029 \def\@tempa{#1}%
3030 \MT@ifstreq\@tempa{character}\relax{%

```

Test whether it's a dimension, but do not translate it into its final form here, since it may be font-specific.

```

3031 \MT@ifdimen\@tempa
3032 {\MT@glet@nc{MT@pr@c@MT@curr@set@name @unit}\@tempa}%
3033 {\MT@warn@nodim{character widths}}%
3034 }%
3035 }

```

Tracking may only be relative to a dimension.

```

3036 \define@key{MT@tr@c}{unit}[1em]{%
3037 \MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@empty
3038 \def\@tempa{#1}%
3039 \MT@ifdimen\@tempa
3040 {\MT@glet@nc{MT@tr@c@MT@curr@set@name @unit}\@tempa}%
3041 {\MT@warn@nodim{1em}}%
3042 \MT@gdef@n{MT@tr@c@MT@curr@set@name @unit}{1em}}%
3043 }

```

Spacing and kerning codes may additionally be relative to space dimensions.

```

3044 \MT@map@clist@n{sp,kn}{%
3045 \define@key{MT@#1@c}{unit}[space]{%
3046 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@empty
3047 \def\@tempa{##1}%
3048 \MT@ifstreq\@tempa{character}\relax{%
3049 \MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\m@ne
3050 \MT@ifstreq\@tempa{space}\relax{%
3051 \MT@ifdimen\@tempa
3052 {\MT@glet@nc{MT@#1@c@MT@curr@set@name @unit}\@tempa}%
3053 {\MT@warn@nodim{width of space}}%
3054 }%
3055 }%
3056 }%
3057 }

```

The first argument to `\SetExpansion` accepts some more options.

```

3058 \MT@map@clist@n{stretch,shrink,step}{%
3059 \define@key{MT@ex@c}{#1}[]{%
3060 \MT@ifempty{##1}\relax{%
3061 \MT@ifint{##1}{%

```

A space terminates the number.

```

3062 \MT@gdef@n{MT@ex@c@MT@curr@set@name @#1}{##1 }%
3063 }{%
3064 \MT@warning{%
3065 Value `##1' for option `#1' is not a number.\MessageBreak
3066 Ignoring it}%
3067 }%
3068 }%
3069 }%
3070 }
3071 \define@key{MT@ex@c}{auto}[true]{%
3072 \def\@tempa{#1}%
3073 \csname if\@tempa\endcsname

```

Don't use autoexpand for pdfTeX version older than 1.20.

```

3074 \MT@requires@pdftex4{%
3075 \MT@gdef@n{MT@ex@c@MT@curr@set@name @auto}{autoexpand}%
3076 }{%
3077 \MT@warning{pdftex too old for automatic font expansion}%
3078 }
3079 \else
3080 \MT@requires@pdftex4{%

```

```

3081 \MT@glet@nc{MT@ex@c@MT@curr@set@name @auto}\@empty
3082 } \relax
3083 \fi
3084 }

```

Tracking: Interword spacing and outer kerning. The variant with space in case \SetTracking is called inside an argument (e. g., to \IfFileExists).

```

3085 \MT@define@opt@key{tr}{spacing}
3086 \MT@define@opt@key{tr}{outerspacing}
3087 \MT@define@opt@key{tr}{outerkerning}

```

Which ligatures should be disabled?

```

3088 \define@key{MT@tr@c}{noligatures}[]%
3089 {\MT@xdef@n{MT@tr@c@MT@curr@set@name @noligatures}{#1}}
3090 \define@key{MT@tr@c}{outer spacing}[]{\setkeys{MT@tr@c}{outerspacing={#1}}}
3091 \define@key{MT@tr@c}{outer kerning}[]{\setkeys{MT@tr@c}{outerkerning={#1}}}
3092 \define@key{MT@tr@c}{no ligatures}[]{\setkeys{MT@tr@c}{noligatures={#1}}}

```

14.3.6 Character inheritance

\DeclareCharacterInheritance This macro may be used in the configuration files to declare characters that should inherit protrusion resp. expansion values from other characters. Thus, there is no need to define all accented characters (e. g., \a, \a, \^a, \~a, \a, \r{a}, \k{a}, \u{a}), which will make the configuration files look much nicer and easier to maintain. If a single character of an inheritance list should have a different value, one can simply override it.

\MT@inh@feat The optional argument may be used to restrict the list to some features,
\MT@extra@inputenc and to specify an input encoding.

```

3093 \renewcommand*\DeclareCharacterInheritance[1][]{%
3094 \let\MT@extra@context\@empty
3095 \let\MT@extra@inputenc\@undefined
3096 \let\MT@inh@feat\@empty
3097 \setkeys{MT@inh@}{#1}%
3098 \MT@begin@catcodes
3099 \MT@set@inh@list
3100 }

```

\MT@set@inh@list Safe category codes.

```

3101 \def\MT@set@inh@list#1#2{%
3102 \MT@ifempty\MT@inh@feat{%
3103 \MT@map@clist@c\MT@features{{\MT@declare@char@inh{##1}{#1}{#2}}}%
3104 }{%
3105 \MT@map@clist@c\MT@inh@feat{%
3106 \KV@sp@def\@tempa{##1}%
3107 \MT@ifempty\@tempa\relax{%
3108 \MT@exp@one@n\MT@declare@char@inh
3109 {\csname MT@rbba@\@tempa\endcsname}{#1}{#2}%
3110 }%
3111 }%
3112 }%
3113 \MT@end@catcodes
3114 }

```

The keys for the optional argument.

```

3115 \MT@map@clist@c\MT@features@long{%
3116 \define@key{MT@inh@}{#1}[]{\edef\MT@inh@feat{\MT@inh@feat#1,}}
3117 \define@key{MT@inh@}{inputenc}{\def\MT@extra@inputenc{#1}}

```

`\MT@declare@char@inh` The lists cannot be given a name by the user.

```

3118 \def\MT@declare@char@inh#1#2#3{%
3119   \MT@edef\MT@#1@inh@name{%
3120     {\MT@curr@file/\the\inputlineno (\@nameuse{MT@abbr@#1})}%
3121     \MT@let\cn\MT@curr@set@name{MT@#1@inh@name}%
3122     \MT@ifdefined\c@T\MT@extra@inputenc{%
3123       \MT@xdef\MT@#1@inh@\MT@curr@set@name @inputenc}{\MT@extra@inputenc}}%
3124   (debug)\MT@edinfo{1}{creating inheritance list ` \@nameuse{MT@#1@inh@name}' }%
3125   \MT@gdef\MT@#1@inh@\csname MT@#1@inh@name\endcsname}{#3}%
3126   \def\MT@permutelist{#1@inh}%
3127   \setkeys{MT@inh}{#2}%
3128   \MT@permute
3129 }
```

Parse the second argument. `\DeclareCharacterInheritance` may also be set up for various combinations.

```

3130 \define@key{MT@inh}{encoding}[]{%
3131   \def\MT@val{#1}%
3132   \expandafter\MT@encoding@check\MT@val,\@nil
3133   \MT@get@highlevel{encoding}%
3134   \MT@edef\MT@tempencoding1{\MT@val}%
3135 }
```

`\MT@encoding@check` But we only allow *one* encoding.

```

3136 \def\MT@encoding@check#1,#2\@nil{%
3137   \MT@ifempty{#2}\relax{%
3138     \edef\MT@val{#1}%
3139     \MT@warning{You may only specify one encoding for character\MessageBreak
3140               inheritance lists. Ignoring encoding(s) #2}%
3141   }%
3142 }
```

For the rest, we can reuse the key setup from the configuration lists (`\Set...`).

```

3143 \MT@define@code@key{family}{inh}
3144 \MT@define@code@key{series}{inh}
3145 \MT@define@code@key{shape}{inh}
3146 \MT@define@code@key{size}{inh}
3147 \MT@define@code@key{font}{inh}
```

`\MT@inh@do` Now parse the third argument, the inheritance lists. We define the commands `\MT@inh@<name>@<slot>`, containing the inheriting characters. They will also be translated to slot numbers here, to save some time. The following will be executed only once, namely the first time this inheritance list is encountered (in `\MT@set@<feature>@codes`).

```

3148 \def\MT@inh@do#1,{%
3149   \ifx\relax#1\@empty \else
3150     \MT@inh@split #1==\relax
3151     \expandafter\MT@inh@do
3152   \fi
3153 }
```

`\MT@inh@split` Only gather the inheriting characters here. Their codes will actually be set in `\MT@set@<feature>@codes`.

```

3154 \def\MT@inh@split#1=#2=#3\relax{%
3155   \def\@tempa{#1}%
3156   \ifx\@tempa\@empty \else
3157     \MT@get@slot
3158     \ifnum\MT@char > \m@ne
3159       \let\MT@val\MT@char
3160       \MT@map@clist\@tempa{#2}%

```

```

3161      \def\@tempa{##1}%
3162      \ifx\@tempa\@empty \else
3163        \MT@get@slot
3164        \ifnum\MT@char > \m@ne
3165          \MT@exp@cs\MT@xadd\MT@inh\MT@listname @\MT@val @{{\MT@char}}%
3166        \fi
3167      \fi
3168    }%
3169    <debug>\MT@info@n1{2}{children of #1 (\MT@val):
3170    <debug>          \@nameuse\MT@inh\MT@listname @\MT@val @}}%
3171    \fi
3172  \fi
3173 }

```

14.3.7 Permutation

\MT@permute Calling \MT@permute will define commands for all permutations of the specified font
 \MT@permute@ attributes of the form \MT@<list type>@/<encoding>/<family>/<series>/<shape>/<|*> to
 \MT@permute@@ be the expansion of \MT@<list type>@name, i. e., the name of the currently defined list.
 \MT@permute@@@ Size ranges are held in a separate macro called \MT@<list type>@/@sizes,
 \MT@permute@@@@ which in turn contains the respective <list name>s attached to the ranges.

```

3174 \def\MT@permute{%
3175   \let\MT@cnt@encoding\@ne
3176   \MT@permute@
3177   \MT@map@tlist@n{{encoding}{family}{series}{shape}}\MT@permute@reset
3178   \MT@gl@t\MT@temp@size\@undefined
3179 }
3180 \def\MT@permute@{%
3181   \let\MT@cnt@family\@ne
3182   \MT@permute@@
3183   \MT@increment\MT@cnt@encoding
3184   \MT@ifdefined@n@T\MT@temp@encoding\MT@cnt@encoding}%
3185   \MT@permute@
3186 }
3187 \def\MT@permute@@{%
3188   \let\MT@cnt@series\@ne
3189   \MT@permute@@@
3190   \MT@increment\MT@cnt@family
3191   \MT@ifdefined@n@T\MT@temp@family\MT@cnt@family}%
3192   \MT@permute@@@
3193 }
3194 \def\MT@permute@@@{%
3195   \let\MT@cnt@shape\@ne
3196   \MT@permute@@@@
3197   \MT@increment\MT@cnt@series
3198   \MT@ifdefined@n@T\MT@temp@series\MT@cnt@series}%
3199   \MT@permute@@@@
3200 }
3201 \def\MT@permute@@@@{%
3202   \MT@permute@@@@@
3203   \MT@increment\MT@cnt@shape
3204   \MT@ifdefined@n@T\MT@temp@shape\MT@cnt@shape}%
3205   \MT@permute@@@@@
3206 }

```

\MT@permute@@@@@ In order to save some memory, we can ignore unused encodings (inside the docu-
 ment).

```

3207 \def\MT@permute@@@@@{%

```

```

3208 \MT@permute@define{encoding}%
3209 \ifMT@document
3210   \ifx\MT@tempencoding\@empty \else
3211     \MT@ifdefined@n@TF{TF\MT@tempencoding}\relax
3212     {\expandafter\expandafter\expandafter\@gobble}%
3213   \fi
3214 \fi
3215 \MT@permute@@@@@
3216 }

```

\MT@permute@@@@@

```

3217 \def\MT@permute@@@@@{%
3218   \MT@permute@define{family}%
3219   \MT@permute@define{series}%
3220   \MT@permute@define{shape}%
3221   \edef\@tempa{\MT@tempencoding
3222     /\MT@tempfamily
3223     /\MT@tempseries
3224     /\MT@tempshape
3225     /\MT@ifdefined@c@T\MT@tempsize *}%

```

Some sanity checks: an encoding must be specified (unless nothing else is).

```

3226 \MT@ifstreq\@tempa{////}\relax{%
3227   \ifx\MT@tempencoding\@empty
3228     \MT@warning{%
3229       You have to specify an encoding for\MessageBreak
3230       \@nameuse{MT@abbr@MT@permutelist} list
3231       ~\@nameuse{MT@MT@permutelist @name}'.\MessageBreak
3232       Ignoring it}%
3233   \else
3234     \MT@ifdefined@c@TF\MT@tempsize{%

```

Add the list of ranges to the beginning of the current combination, after checking for conflicts.

```

3235   \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context @sizes}{%
3236     \MT@map@tlist@c@MT@tempsize\MT@check@rlist
3237   }%
3238   \MT@exp@cs\MT@xaddb
3239   {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3240   \MT@tempsize
3241   <debug>\MT@info@n1{1}{initialising: use list for font \@tempa,\MessageBreak
3242   <debug>       sizes: \csname MT@MT@permutelist @\@tempa\MT@extra@context
3243   <debug>       @sizes\endcsname}%
3244   }{%

```

Only one list can apply to a given combination.

```

3245   \MT@ifdefined@n@T{MT@MT@permutelist @\@tempa\MT@extra@context}{%
3246     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3247       ~\@nameuse{MT@MT@permutelist @name}' will override list\MessageBreak
3248       ~\@nameuse{MT@MT@permutelist @\@tempa\MT@extra@context}'
3249       for font \@tempa'}%
3250   }%
3251   <debug>\MT@info@n1{1}{initialising: use list for font \@tempa
3252   <debug>       \ifx\MT@extra@context\@empty\else\MessageBreak
3253   <debug>       (context: \MT@extra@context)\fi}%
3254   }%
3255   \MT@xdefn{MT@MT@permutelist @\@tempa\MT@extra@context}%
3256   {\csname MT@MT@permutelist @name\endcsname}%
3257   \fi
3258   }%
3259 }

```

`\MT@permute@define` Define the commands.

```
3260 \def\MT@permute@define#1{%
3261   \@tempcnta=\csname MT@cnt@#1\endcsname\relax
3262   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3263   {\MT@edef@n{MT@temp#1}{\csname MT@temp#1\the\@tempcnta\endcsname}}%
3264   {\MT@let@nc{MT@temp#1}\@empty}%
3265 }
```

`\MT@permute@reset` Reset the commands.

```
3266 \def\MT@permute@reset#1{%
3267   \@tempcnta=\@ne
3268   \MT@loop
3269   \MT@let@nc{MT@temp#1\the\@tempcnta}\@undefined
3270   \advance\@tempcnta\@ne
3271   \MT@ifdefined@n@TF{MT@temp#1\the\@tempcnta}%
3272   \iftrue
3273   \iffalse
3274   \MT@repeat
3275 }
```

`\MT@check@rlist` For every new range item in `\MT@tempsize`, check whether it overlaps with ranges in the existing list.

```
3276 \def\MT@check@rlist#1{\expandafter\MT@check@rlist@ #1}
```

`\MT@check@rlist@` Define the current new range and ...

```
3277 \def\MT@check@rlist@#1#2#3{%
3278   \def\@tempb{#1}%
3279   \def\@tempc{#2}%
3280   \MT@if@false
3281   \MT@exp@cs\MT@map@tlist@c
3282   {MT@MT@permutelist @\@tempa\MT@extra@context @sizes}%
3283   \MT@check@range
3284 }
```

`\MT@check@range` ... recurse through the list of existing ranges.

```
3285 \def\MT@check@range#1{\expandafter\MT@check@range@ #1}
```

`\MT@check@range@` `\@tempb` and `\@tempc` are lower resp. upper bound of the new range, `<#2>` and `<#3>` those of the existing range.

```
3286 \def\MT@check@range@#1#2#3{%
3287   \MT@ifdim{#2}=\m@ne{%
3288     \MT@ifdim\@tempc=\m@ne{%
```

- Both items are simple sizes.

```
3289     \MT@ifdim\@tempb={#1}\MT@if@true\relax
3290   }{%
```

- Item in list is a simple size, new item is a range.

```
3291     \MT@ifdim\@tempb>{#1}\relax{%
3292     \MT@ifdim\@tempc>{#1}%
3293     \MT@if@true
3294     \edef\@tempb{#1 (with range: \@tempb\space to \@tempc)}%
3295   }\relax
3296 }%
3297 }%
3298 }{%
3299   \MT@ifdim\@tempc=\m@ne{%
```

- Item in list is a range, new item is a simple size.

```

3300     \MT@ifdim\@tempb<{#2}{%
3301         \MT@ifdim\@tempb<{#1}\relax\MT@iftrue
3302     }\relax
3303 }{%

```

- Both items are ranges.

```

3304     \MT@ifdim\@tempb<{#2}{%
3305         \MT@ifdim\@tempc>{#1}{%
3306             \MT@iftrue
3307             \edef\@tempb{#1 to #2 (with range: \@tempb\space to \@tempc)}%
3308         }\relax
3309     }\relax
3310 }%
3311 }%
3312 \ifMT@if@
3313     \MT@warning{\@nameuse{MT@abbr@MT@permutelist} list
3314         ~\@nameuse{MT@MT@permutelist @name}' will override\MessageBreak
3315         list ~#3' for font \@tempa,\MessageBreak size \@tempb}%

```

If we've already found a conflict with this item, we can skip the rest of the list.

```

3316     \expandafter\MT@tlist@break
3317     \fi
3318 }

```

14.4 Package options

14.4.1 Declaring the options

`\ifMT@opt@expansion` Keep track of whether the user explicitly set these options.

```

\ifMT@opt@auto 3319 \newif\ifMT@opt@expansion
\ifMT@opt@DVI 3320 \newif\ifMT@opt@auto
                 3321 \newif\ifMT@opt@DVI

```

`\MT@optwarn@admissible` Some warnings.

```

3322 \def\MT@optwarn@admissible#1#2{%
3323     \MT@warning@nl{`#1' is not an admissible value for option\MessageBreak
3324         ~#2'. Assuming `false'}%
3325 }

```

`\MT@optwarn@nan`

```

3326 </package>
3327 <plain>\MT@requires@latex1{
3328     \def\MT@optwarn@nan#1#2{%
3329         \MT@warning@nl{Value ~#1' for option ~#2' is not a\MessageBreak number.
3330             Using default value of \number\@nameuse{MT@#2@default}}%
3331     }
3332 <plain>}\relax
3333 <*package>

```

`\MT@opt@def@set`

```

3334 \def\MT@opt@def@set#1{%
3335     \MT@ifdefined@n@TF{MT@\@tempb @set@@\MT@val}{%
3336         \MT@xdef@n{MT@\@tempb @setname}{\MT@val}%
3337     }{%
3338         \MT@xdef@n{MT@\@tempb @setname}{\@nameuse{MT@default@\@tempb @set}}%
3339         \MT@warning@nl{The #1 set ~\MT@val' is undeclared.\MessageBreak
3340             Using set ~\@nameuse{MT@\@tempb @setname}' instead}%
3341     }%
3342 }

```


expansion and protrusion may be true, false, compatibility, nocompatibility and/or a *set name*).

```

3343 \MT@map@clist@n{protrusion,expansion}{%
3344   \define@key{MT}{#1}[true]{%
3345     \csname MT@opt@#1true\endcsname
3346     \MT@map@clist@n{##1}{%
3347       \KV@sp@def\MT@val{###1}%
3348       \MT@ifempty\MT@val\relax{%
3349         \csname MT@#1true\endcsname
3350         \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3351         \MT@ifstreq\MT@val{true}\relax
3352         {%
3353           \MT@ifstreq\MT@val{false}{%
3354             \csname MT@#1false\endcsname
3355             }{%
3356               \MT@ifstreq\MT@val{compatibility}{%
3357                 \MT@let@nc{MT@\@tempb @level}\@ne
3358                 }{%
3359                   \MT@ifstreq\MT@val{nocompatibility}{%
3360                     \MT@let@nc{MT@\@tempb @level}\tw@
3361                     }{%

```

If everything failed, it should be a set name.

```

3362       \MT@opt@def@set{#1}%
3363     }%
3364   }%
3365 }%
3366 }%
3367 }%
3368 }%
3369 }%
3370 }

```

activate is a shortcut for protrusion and expansion.

```

3371 \define@key{MT}{activate}[true]{%
3372   \setkeys{MT}{protrusion={#1}}%
3373   \setkeys{MT}{expansion={#1}}%
3374 }

```

spacing, kerning and tracking do not have a compatibility level.

```

3375 \MT@map@clist@n{spacing,kerning,tracking}{%
3376   \define@key{MT}{#1}[true]{%
3377     \MT@map@clist@n{##1}{%
3378       \KV@sp@def\MT@val{###1}%
3379       \MT@ifempty\MT@val\relax{%
3380         \csname MT@#1true\endcsname
3381         \MT@ifstreq\MT@val{true}\relax
3382         {%
3383           \MT@ifstreq\MT@val{false}{%
3384             \csname MT@#1false\endcsname
3385             }{%
3386               \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3387               \MT@opt@def@set{#1}%
3388               }%
3389             }%
3390           }%
3391         }%
3392       }%
3393     }

```

\MT@def@bool@opt The true/false options: draft, final (may be inherited from the class options),

auto, selected, babel, DVInoutput, defersetup, copyfonts.

```

3394 \def\MT@def@bool@opt#1#2{%
3395   \define@key{MT}{#1}[true]{%
3396     \def\@tempa{##1}%
3397     \MT@ifstreq\@tempa{true}\relax{%
3398       \MT@ifstreq\@tempa{false}\relax{%
3399         \MT@optwarn@admissible{##1}{#1}%
3400         \def\@tempa{false}%
3401       }%
3402     }%
3403     #2%
3404   }%
3405 }

```

Boolean options that only set the switch.

```

3406 \MT@map@clist@n{draft,selected,babel}{%
3407   \MT@def@bool@opt{#1}{\csname MT@#1\@tempa\endcsname}}
3408 \MT@def@bool@opt{auto}{\csname MT@auto\@tempa\endcsname \MT@opt@autotruer}

```

The DVInoutput option will change \pdfoutput immediately to minimise the risk of confusing other packages.

```

3409 \MT@def@bool@opt{DVInoutput}{%
3410   \csname if\@tempa\endcsname
3411     \ifnum\pdfoutput>\z@ \MT@opt@DVIttrue \fi
3412     \pdfoutput\z@
3413   \else
3414     \ifnum\pdfoutput<\@ne \MT@opt@DVIttrue \fi
3415     \pdfoutput\@ne
3416   \fi
3417 }

```

Setting the defersetup option to false will restore the old behaviour, where the setup took place at the time when the package was loaded. This is undocumented, since I would like to learn about the cases where this is necessary.

The only problem with the new deferred setup I can think of is when a box is being constructed inside the preamble and this box contains a font that is not loaded before the box is being used.

```

3418 \MT@def@bool@opt{defersetup}{%
3419   \csname if\@tempa\endcsname \else
3420     \AtEndOfPackage{%
3421       \MT@setup@
3422       \let\MT@setup@\@empty
3423       \let\MT@addto@setup\@firstofone
3424     }%
3425   \fi
3426 }

```

copyfonts will copy all fonts before setting them up. This allows protrusion and expansion with different parameters. This options is also *undocumented* in the hope that we can always find out automatically whether it's required.

```

3427 \MT@requires@pdftex{
3428   (*lua)
3429   \MT@requires@lualatex{
3430     \MT@def@bool@opt{copyfonts}{%
3431       \csname if\@tempa\endcsname
3432         \MT@error{The `copyfonts' option doesn't work with lualatex}
3433         {Use pdftex instead.}%
3434       \fi
3435     }
3436   }{

```

```

3437 </lua>
3438 \MT@def@bool@opt{copyfonts}{%
3439   \csname if\@tempa\endcsname
3440     \MT@glet\MT@copy@font\MT@copy@font@
3441   \else
3442     \MT@glet\MT@copy@font\relax
3443   \fi
3444 }
3445 <lua> }
3446 }{
3447 \MT@def@bool@opt{copyfonts}{%
3448   \csname if\@tempa\endcsname
3449     \MT@error{The pdftex version you are using is too old\MessageBreak
3450       to use the 'copyfonts' option}{Upgrade pdftex.}%
3451   \fi
3452 }
3453 }

```

`final` is the opposite to `draft`.

```

3454 \MT@def@bool@opt{final}{%
3455   \csname if\@tempa\endcsname
3456     \MT@draftfalse
3457   \else
3458     \MT@drafttrue
3459   \fi
3460 }

```

For verbose output, we redefine `\MT@vinfo`.

```

3461 \define@key{MT}{verbose}[true]{%
3462   \let\MT@vinfo\MT@info@n1
3463   \def\@tempa{#1}%
3464   \MT@ifstreq\@tempa{true}\relax{%

```

Take problems seriously.

```

3465   \MT@ifstreq\@tempa{errors}{%
3466     \let\MT@warning \MT@warn@err
3467     \let\MT@warning@n1\MT@warn@err
3468   }{%
3469     \let\MT@vinfo\@gobble

```

Cast warnings to the winds.

```

3470   \MT@ifstreq\@tempa{silent}{%
3471     \let\MT@warning \MT@info
3472     \let\MT@warning@n1\MT@info@n1
3473   }{%
3474     \MT@ifstreq\@tempa{false}\relax{\MT@optwarn@admissible{#1}{verbose}}%
3475   }%
3476 }%
3477 }%
3478 }

```

Options with numerical keys: `factor`, `stretch`, `shrink`, `step`, `letterspace`.

```

3479 </package>
3480 <plain>\MT@requires@latex1{
3481 \MT@map@clist@n{%
3482 <package> stretch,shrink,step,%
3483   letterspace}{%
3484   \define@key{MT}{#1}[\csname MT@#1@default\endcsname]{%
3485     \def\@tempa{##1 }%

```

No nonsense in `\MT@factor` et al.? A space terminates the number.

```

3486   \MT@ifint\@tempa
3487   {\MT@edef\@tempa{#1}\@tempa}%

```

```

3488     {\MT@optwarn@nan{##1}{#1}}%
3489   }%
3490 }
3491 \plain\relax
3492 \*package

```

factor will define the protrusion factor only.

```

3493 \define@key{MT}{factor}{\MT@factor@default}{%
3494   \def\@tempa{#1}%
3495   \MT@ifint\@tempa
3496   {\edef\MT@pr@factor{\@tempa}}
3497   {\MT@optwarn@nan{#1}{factor}}}%
3498 }

```

Unit for protrusion codes.

```

3499 \define@key{MT}{unit}{character}{%
3500   \def\@tempa{#1}%
3501   \MT@ifstreq\@tempa{character}\relax{%
3502     \MT@ifdimen\@tempa
3503     {\let\MT@pr@unit\@tempa}%
3504     {\MT@warning@nl{\@tempa' is not a dimension.\MessageBreak
3505       Ignoring it and setting values relative to\MessageBreak
3506       character widths}}}%
3507   }%
3508 }

```

14.4.2 Reading the configuration file

The package should just work if called without any options. Therefore, expansion will be switched off by default if output is DVI, since it isn't likely that expanded fonts are available. (This grows more important as modern \TeX systems have switched to the pdf \TeX engine even for DVI output, so that the user might not even be aware of the fact that she's running pdf \TeX .)

```

3509 \MT@protrusiontrue
3510 \ifnum\pdfoutput<\@ne \else

```

Also, we only enable expansion by default if pdf \TeX can expand the fonts automatically.

```

3511   \MT@requirespdfTeX4{
3512     \MT@expansiontrue
3513     \MT@autottrue
3514   }\relax
3515 \fi

```

The main configuration file will be loaded before processing the package options.

`\MT@config@file` However, the config option must of course be evaluated beforehand. We also have to define a no-op for the regular option processing later.

```

\MT@get@config
3516 \define@key{MT}{config}[]{\relax}
3517 \def\MT@get@config#1config=#2,#3\@nil{%
3518   \MT@ifempty{#2}%
3519   {\def\MT@config@file{\MT@MT.cfg}}%
3520   {\def\MT@config@file{#2.cfg}}%
3521 }
3522 \expandafter\expandafter\expandafter\MT@get@config
3523 \csname opt@\currname.\@currxt\endcsname,config=\@nil

```

Load the file.

```

3524 \IfFileExists{\MT@config@file}{%
3525   \MT@info@nl{Loading configuration file \MT@config@file}%

```

```

3526 \MT@begin@catcodes
3527 \let\MT@begin@catcodes\relax
3528 \let\MT@end@catcodes\relax
3529 \let\MT@curr@file\MT@config@file
3530 \input{\MT@config@file}%
3531 \endgroup
3532 }{\MT@warning@nl{%
3533   Could not find configuration file `~\MT@config@file'!\MessageBreak
3534   This will almost certainly cause undesired results.\MessageBreak
3535   Please fix your installation}%
3536 }

```

`\MT@check@active@set` We have to make sure that font sets are active. If the user didn't activate any, we use those sets declared by `\DeclareMicrotypeSetDefault` (this is done at the end of the preamble).

```

3537 \def\MT@check@active@set#1{%
3538   \MT@ifdefined@n@TF{MT@#1@setname}{%
3539     \MT@info@nl{Using \@nameuse{MT@abbr@#1} set `~\@nameuse{MT@#1@setname}'}%
3540   }{%
3541     \MT@ifdefined@n@TF{MT@default@#1@set}{%
3542       \MT@gl@et@nn{MT@#1@setname}{MT@default@#1@set}%
3543       \MT@info@nl{Using default \@nameuse{MT@abbr@#1} set `~\@nameuse{MT@#1@setname}'}%
3544     }{%

```

If no default font set has been declared in the main configuration file, we use the (empty, non-existent) set '@', and issue a warning.

```

3545       \MT@gdef@n{MT@#1@setname}{@}%
3546       \MT@warning@nl{No \@nameuse{MT@abbr@#1} set chosen, no default set declared.
3547         \MessageBreak Using empty set}%
3548     }%
3549   }%
3550 }

```

14.4.3 Hook for other packages

`\Microtype@Hook` This hook may be used by font package authors, e. g., to declare alias fonts. If it is defined, it will be executed here, i. e., after the main configuration file has been loaded, and before the package options are evaluated.

This hook was needed in versions prior to 1.9a to overcome the situation that (1) the microtype package should be loaded after all font defaults have been set up (hence, using `\ifpackageloaded` in the font package was not viable), and (2) checking `\AtBeginDocument` could be too late, since fonts might already have been loaded, and consequently set up, in the preamble. With the new deferred setup, one could live without this command, however, it remains here since it's simpler than testing whether the package was loaded both in the preamble as well as at the beginning of the document (which is what one would have to do).

Package authors should check whether the command is already defined so that existing definitions by other packages aren't overwritten. Example:

```

\def\MinionPro@MT@Hook{\DeclareMicrotypeAlias{MinionPro-LF}{MinionPro}}
\ifpackageloaded{microtype}
  \MinionPro@MT@Hook
  {\@ifundefined{Microtype@Hook}
    {\let\Microtype@Hook\MinionPro@MT@Hook
     {\gaddto@macro\Microtype@Hook{\MinionPro@MT@Hook}}}

```

\MicroType@Hook with a capital T (which only existed in version 1.7) is provided for compatibility reasons. At some point in the future, it will no longer be available, hence it should not be used.

```
3551 \MT@ifdefined@c@T\MicroType@Hook{\MT@warning{%
3552   Command \string\MicroType@Hook\space is deprecated.\MessageBreak
3553   Use \string\Microtype@Hook\space instead}\MicroType@Hook}
3554 \MT@ifdefined@c@T\Microtype@Hook\Microtype@Hook
```

14.4.4 Changing options later

\microtypesetup Inside the preamble, \microtypesetup accepts the same options as the package (un-
\MT@define@optionX less defersetup=false). In the document body, it accepts the options: protrusion,
expansion, activate, tracking, spacing and kerning. Specifying font sets is not
allowed.

```
3555 \def\microtypesetup{\setkeys{MT}}
3556 \MT@addto@setup{\def\microtypesetup#1{\setkeys{MTX}{#1}\selectfont}}
3557 \def\MT@define@optionX#1#2{%
3558   \define@key{MTX}{#1}[true]{%
3559     \edef\@tempb{\csname MT@rbba@#1\endcsname}%
3560     \MT@map@clist@{\@tempb}%
3561     \KV@sp@def\MT@val{###1}%
3562     \MT@ifempty\MT@val\relax{%
3563       \@tempcnta=\m@ne
3564       \MT@ifstreq\MT@val{true}{%
3565         \MT@checksetup{#1}{%
3566           \@tempcnta=\csname MT@\@tempb @level\endcsname
3567           \MT@vinfo{Enabling #1
3568             (level \number\csname MT@\@tempb @level\endcsname)\on@line}%
3569           }%
3570         }%
3571         \MT@ifstreq\MT@val{false}{%
3572           \@tempcnta=\z@
3573           \MT@vinfo{Disabling #1\on@line}%
3574         }%
3575         \MT@ifstreq\MT@val{compatibility}{%
3576           \MT@checksetup{#1}{%
3577             \@tempcnta=\@ne
3578             \MT@let@nc{MT@\@tempb @level}\@ne
3579             \MT@vinfo{Setting #1 to level 1\on@line}%
3580           }%
3581         }%
3582         \MT@ifstreq\MT@val{nocompatibility}{%
3583           \MT@checksetup{#1}{%
3584             \@tempcnta=\tw@
3585             \MT@let@nc{MT@\@tempb @level}\tw@
3586             \MT@vinfo{Setting #1 to level 2\on@line}%
3587           }%
3588         }%
3589         \MT@error{Value '\MT@val' for key '#1' not recognised}
3590         {Use any of 'true', 'false', 'compatibility' or
3591           'nocompatibility'.}%
3592       }%
3593     }%
3594   }%
3595   \ifnum\@tempcnta>\m@ne
```

Enabling micro-typography in the middle of the document is not allowed if it has been disabled in the package options since fonts might already have been loaded and hence wouldn't be set up.

```

3596         #2\@tempcnta\relax
3597     \fi
3598 }%
3599 }%
3600 }%
3601 }

\MT@checksetup    Test whether the feature wasn't disabled in the package options.
3602 \def\MT@checksetup#1{%
3603     \csname ifMT@#1\endcsname
3604     \expandafter\@firstofone
3605     \else
3606     \MT@error{You cannot enable #1 if it was disabled\MessageBreak
3607         in the package options}{Load microtype with #1 enabled.}%
3608     \expandafter\@gobble
3609     \fi
3610 }

3611 \MT@define@optionX{protrusion}\pdfprotrudechars
3612 \MT@define@optionX{expansion}\pdfadjustspacing

\MT@define@optionX@    The same for tracking, spacing and kerning, which do not have a compatibility
                        level.
3613 \MT@requires@pdftex6{
3614     \lua \MT@requires@luatex\@firstofone{
3615         \def\MT@define@optionX@#1#2{%
3616             \define@key{MTX}{#1}[true]{%
3617                 \MT@map@clist@n{##1}{%
3618                     \KV@sp@def\MT@val{###1}%
3619                     \MT@iifempty\MT@val\relax{%
3620                         \@tempcnta=\m@ne
3621                         \MT@ifstreq\MT@val{true}{%
3622                             \MT@checksetup{#1}{%
3623                                 \@tempcnta=\@ne
3624                                 \MT@vinfo{Enabling #1\on@line}%
3625                             }%
3626                         }{%
3627                             \MT@ifstreq\MT@val{false}{%
3628                                 \@tempcnta=\z@
3629                                 \MT@vinfo{Disabling #1\on@line}%
3630                             }{\MT@error{Value '\MT@val' for key '#1' not recognised}
3631                                 {Use either 'true' or 'false'}}%
3632                         }%
3633                     }%
3634                     \ifnum\@tempcnta>\m@ne
3635                         #2\relax
3636                     \fi
3637                 }%
3638             }%
3639         }%
3640     }

                        We cannot simply let \MT@tracking relax, since this may select the already letter-
                        spaced font instance.
3641     \MT@define@optionX@{tracking}{\ifnum\@tempcnta=\z@ \let\MT@tracking\MT@set@tr@zero
3642         \else \let\MT@tracking\MT@tracking@ \fi}
3643     \MT@define@optionX@{spacing}{\pdfadjustinterwordglue\@tempcnta}
3644     \MT@define@optionX@{kerning}{\pdfprependkern\@tempcnta
3645         \pdfappendkern \@tempcnta}
3646     \@gobble
3647     \lua }
3648 } \@firstofone

```

Disable for older pdfTeX versions and for luaTeX.

```

3649 {\define@key{MTX}{tracking}[true]{\MT@warning{Ignoring tracking setup}}
3650 \define@key{MTX}{kerning}[true]{\MT@warning{Ignoring kerning setup}}
3651 \define@key{MTX}{spacing}[true]{\MT@warning{Ignoring spacing setup}}
3652 }
3653 \define@key{MTX}{activate}[true]{%
3654   \setkeys{MTX}{protrusion={#1}}%
3655   \setkeys{MTX}{expansion={#1}}%
3656 }

```

`\MT@saved@setupfont` Disable everything – may be used as a work-around in case setting up fonts doesn't work in certain environments. (*Undocumented.*)

```

3657 \let\MT@saved@setupfont\MT@setupfont
3658 \define@key{MTX}{disable}[]{%
3659   \MT@info{Inactivate `~\MT@MT' package}%
3660   \let\MT@setupfont\relax
3661 }
3662 \define@key{MTX}{enable}[]{%
3663   \MT@info{Reactivate `~\MT@MT' package}%
3664   \let\MT@setupfont\MT@saved@setupfont
3665 }
3666 </package>

```

14.4.5 Processing the options

`\MT@ProcessOptionsWithKV` Parse options.

```

3667 <plain>\MT@requires@latex1{
3668 \def\MT@ProcessOptionsWithKV#1{%
3669   \let\@tempc\relax
3670   \let\MT@temp\@empty
3671 <plain> \MT@requires@latex2{
3672   \MT@map@clist@c\@classoptionslist{%
3673     \def\CurrentOption{##1}%
3674     \MT@ifdefined@n@T{KV@#1@}\expandafter\MT@getkey\CurrentOption=\@nil}{%
3675       \edef\MT@temp{\MT@temp,\CurrentOption,}%
3676       \@expandtwoargs\@removeelement\CurrentOption
3677       \@unusedoptionlist\@unusedoptionlist
3678     }%
3679   }%
3680   \edef\MT@temp{\noexpand\setkeys{#1}%
3681     {\MT@temp\@optionlist{\@currname.\@currentx}}}%

```

`plain` can handle package options.

```

3682 <*plain>
3683   {\edef\MT@temp{\noexpand\setkeys{#1}%
3684     {\csname usepkg@options@usepkg@pkg\endcsname}}}
3685 </plain>
3686   \MT@temp
3687   \MT@clear@options
3688 }

```

`\MT@getkey` For key=val in class options.

```

3689 \def\MT@getkey#1=#2\@nil{#1}
3690 \MT@ProcessOptionsWithKV{MT}
3691 <plain>}\relax
3692 <*package>

```

Now we can take the appropriate actions. We also tell the log file which options the user has chosen (in case it's interested).


```

3693 \MT@addto@setup{
3694 \ifMT@draft

```

We disable most of what we've just defined in the 3694 lines above if we are running in draft mode.

```

3695 \MT@warning@n1{\`draft' option active.\MessageBreak
3696             Disabling all micro-typographic extensions.\MessageBreak
3697             This might lead to different line and page breaks}
3698 \let\MT@setupfont\relax
3699 \renewcommand*\LoadMicrotypeFile[1]{}
3700 \renewcommand*\microtypesetup[1]{}
3701 \renewcommand*\microtypecontext[1]{}
3702 \renewcommand*\lstyle{}
3703 \else

```

For DVI output, the user must have explicitly passed the expansion option to the package.

```

3704 \ifnum\pdfoutput<\@ne
3705 \ifMT@opt@expansion \else
3706 \MT@expansionfalse
3707 \fi
3708 \fi

```

pdf_{TeX} can create DVI output, too. However, both the DVI viewer and dvips need to find actual fonts. Therefore, expansion will only work if the fonts for different degrees of expansion are readily available.

Some packages depend on the value of `\pdfoutput` and will get confused if it is changed after they have been loaded. These packages are, among others: `color`, `graphics`, `hyperref`, `crop`, `contour`, `pstricks` and, as a matter of course, `ifpdf`. Instead of testing for each package (that's not our job), we only say that it was microtype that changed it. This must be sufficient!

```

3709 \MT@info@n1{Generating \ifnum\pdfoutput<\@ne DVI \else PDF \fi output%
3710             \ifMT@opt@DVI\space (changed by \MT@MT)\fi}%

```

Working on font copies?

```

3711 \ifx\MT@copy@font\relax\else \MT@info@n1{Using font copies for contexts}\fi

```

Fix the font sets.

```

3712 \MT@map@tlist@c\MT@font@sets\MT@fix@font@set

```

Protrusion.

```

3713 \ifMT@protrusion
3714 \edef\MT@active@features{\MT@active@features,pr}
3715 \pdfprotrudechars\MT@pr@level
3716 \MT@info@n1{Character protrusion enabled (level \number\MT@pr@level)}%
3717 \ifnum\MT@pr@factor=\MT@factor@default \else,\MessageBreak
3718     factor: \number\MT@pr@factor\fi
3719 \ifx\MT@pr@unit\@empty \else,\MessageBreak unit: \MT@pr@unit\fi}
3720 \MT@check@active@set{pr}
3721 \else
3722 \let\MT@protrusion\relax
3723 \MT@info@n1{No character protrusion}
3724 \fi

```

Expansion.

```

3725 \ifMT@expansion

```

Set up the values for font expansion: if `stretch` has not been specified, we take the default value of 20.

```

3726 \ifnum\MT@stretch=\m@ne
3727 \let\MT@stretch\MT@stretch@default
3728 \fi

```

If `shrink` has not been specified, it will inherit the value from `stretch`.

```

3729 \ifnum\MT@shrink=\m@ne
3730 \let\MT@shrink\MT@stretch
3731 \fi

```

If `step` has not been specified, we will just set it to 1 for recent pdfTeX versions. My tests did not show much difference neither in compilation time (within the margin of error) nor in file size (less than 1% difference for `microtype.pdf` with `step=1` compared to `step=5`). With older versions, we set it to $\min(\text{stretch}, \text{shrink})/5$, rounded off, minimum value 1.

```

3732 \MT@requires@pdfTeX6{\def\MT@step{1 }}{
3733 \ifnum\MT@step=\m@ne
3734 \ifnum\MT@stretch>\MT@shrink
3735 \ifnum\MT@shrink=\z@
3736 \@tempcnta=\MT@stretch
3737 \else
3738 \@tempcnta=\MT@shrink
3739 \fi
3740 \else
3741 \ifnum\MT@stretch=\z@
3742 \@tempcnta=\MT@shrink
3743 \else
3744 \@tempcnta=\MT@stretch
3745 \fi
3746 \fi
3747 \divide\@tempcnta 5\relax
3748 \else
3749 \@tempcnta=\MT@step
3750 \ifnum\@tempcnta=\z@
3751 \MT@warning@n1{The expansion step cannot be set to zero.\MessageBreak
3752 Setting it to one}
3753 \fi
3754 \fi
3755 \ifnum\@tempcnta=\z@ \@tempcnta=\@ne \fi
3756 \edef\MT@step{\number\@tempcnta\space}}

```

`\MT@auto` Automatic expansion of the font? This new feature of pdfTeX 1.20 makes the *l^hz* programme really usable. It must be either ‘autoexpand’ or empty (or ‘1000’ for older versions of pdfTeX).

```

3757 \let\MT@auto\empty
3758 \ifMT@auto
3759 \MT@requires@pdfTeX4{%

```

We turn off automatic expansion if output mode is DVI.

```

3760 \ifnum\pdfoutput<\@ne
3761 \ifMT@opt@auto
3762 \MT@error{%
3763 Automatic font expansion only works for PDF output.\MessageBreak
3764 However, you are creating a DVI file}
3765 {If you have created expanded fonts instances, remove ‘auto’ from%
3766 \MessageBreak the package options. Otherwise, you have to switch
3767 off expansion.\MessageBreak completely.}
3768 \fi
3769 \MT@autofalse
3770 \else
3771 \def\MT@auto{autoexpand}
3772 \fi

```

Also, if pdfTeX is too old.

```

3773     }{%
3774     \MT@error{%
3775     The pdftex version you are using is too old for\MessageBreak
3776     automatic font expansion}%
3777     {If you have created expanded fonts instances, remove 'auto' from\MessageBreak
3778     the package options. Otherwise, you have to switch off expansion\MessageBreak
3779     completely, or upgrade pdftex to version 1.20 or newer.}
3780     \MT@autofalse
3781     \def\MT@auto{1000 }
3782     }
3783 \else

```

No automatic expansion.

```

3784     \MT@requires@pdftex4\relax{
3785     \def\MT@auto{1000 }
3786     }
3787 \fi

```

Choose the appropriate macro for selected expansion.

```

3788 \ifMT@selected
3789 \let\MT@set@ex@codes\MT@set@ex@codes@s
3790 \else
3791 \let\MT@set@ex@codes\MT@set@ex@codes@n
3792 \fi

```

Filter out stretch=0, shrink=0, since it would result in a pdfTeX error.

```

3793 \ifnum\MT@stretch=\z@
3794 \ifnum\MT@shrink=\z@
3795 \MT@warning@nl{%
3796 Both the stretch and shrink limit are set to zero.\MessageBreak
3797 Disabling font expansion}
3798 \MT@expansionfalse
3799 \fi
3800 \fi
3801 \fi
3802 \ifMT@expansion
3803 \edef\MT@active@features{\MT@active@features,ex}%
3804 \pdfadjustspacing\MT@ex@level
3805 \MT@info@nl{\ifMT@auto A\else Non-a\fi utomatic font expansion enabled
3806 (level \number\MT@ex@level),\MessageBreak
3807 stretch: \number\MT@stretch, shrink: \number\MT@shrink,
3808 step: \number\MT@step, \ifMT@selected\else non-\fi selected}

```

\MT@check@step Check whether stretch and shrink are multiples of step.

```

3809 \def\MT@check@step#1{%
3810 \@tempcnta=\csname MT@#1\endcsname
3811 \divide\@tempcnta \MT@step
3812 \multiply\@tempcnta \MT@step
3813 \ifnum\@tempcnta=\csname MT@#1\endcsname\else
3814 \MT@warning@nl{The #1 amount is not a multiple of step.\MessageBreak
3815 The effective maximum #1 is \the\@tempcnta\space
3816 (step \number\MT@step)}
3817 \fi
3818 }
3819 \MT@check@step{stretch}
3820 \MT@check@step{shrink}
3821 \MT@check@active@set{ex}

```

Inside \showhyphens, font expansion should be disabled.

```

3822 \CheckCommand*\showhyphens[1]{\setbox0\vbox{%
3823 \color@begingroup\everypar{}\parfillskip\z@skip

```

```

3824      \hsize\maxdimen\normal font\pretolerance\m@ne\tolerance\m@ne
3825      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

\showhyphens    I wonder why it's defined globally (in ltfsbas.dtx)?

3826      \gdef\showhyphens#1{\setbox0\vbox{%
3827      \color@begingroup\pdfadjustspacing\z@\everypar{}\parfillskip\z@skip
3828      \hsize\maxdimen\normal font\pretolerance\m@ne\tolerance\m@ne
3829      \hbadness\z@\showboxdepth\z@\ #1\color@endgroup}}

3830      \else
3831      \let\MT@expansion\relax
3832      \MT@info@n1{No font expansion}
3833      \fi
3834      }
3835      \MT@requires@pdftex6{

\MT@warn@lua

3836      <lua>
3837      \def\MT@warn@lua#1{%
3838      \MT@error{The `#1' feature doesn't currently work\MessageBreak with luatex}
3839      {Use pdftex instead.}%
3840      \MT@let@nc{MT@#1}\relax
3841      }
3842      </lua>

3843      </package>
3844      \MT@addto@setup{%
3845      <package>

```

Tracking, spacing and kerning.

```

3846      \ifMT@tracking
3847      <lua>      \MT@requires@luatex{\MT@warn@lua{tracking}}{
3848      \edef\MT@active@features{\MT@active@features,tr}
3849      \MT@info@n1{Tracking enabled}
3850      \MT@check@active@set{tr}

Enable protrusion for compensation at the line edges.

3851      \ifMT@protrusion\else\pdfprotrudechars\@ne\fi
3852      <lua>      }
3853      \else
3854      \let\MT@tracking\relax
3855      \MT@info@n1{No tracking}
3856      \fi
3857      \ifMT@spacing
3858      <lua>      \MT@requires@luatex{\MT@warn@lua{spacing}}{
3859      \edef\MT@active@features{\MT@active@features,sp}
3860      \pdfadjustinterwordglue\@ne
3861      \MT@info@n1{Adjustment of interword spacing enabled}
3862      \MT@check@active@set{sp}
3863      <lua>      }
3864      \else
3865      \let\MT@spacing\relax
3866      \MT@info@n1{No adjustment of interword spacing}
3867      \fi
3868      \ifMT@kerning
3869      <lua>      \MT@requires@luatex{\MT@warn@lua{kerning}}{
3870      \edef\MT@active@features{\MT@active@features,kn}
3871      \pdfprependkern\@ne
3872      \pdfappendkern\@ne
3873      \MT@info@n1{Adjustment of character kerning enabled}
3874      \MT@check@active@set{kn}
3875      <lua>      }
3876      \else

```

```

3877     \let\MT@kerning\relax
3878     \MT@info@nl{No adjustment of character kerning}
3879     \fi
3880 </package>

```

\MT@warn@tracking@DVI We issue a warning, when letterspacing in DVI mode, since it will probably not work. We also switch on protrusion if it isn't already, to compensate for the letterspacing kerns.

```

3881     \ifnum\pdfoutput<\@ne
3882     \def\MT@warn@tracking@DVI{%
3883         \MT@warning@nl{%
3884             You are using tracking/letterspacing in DVI mode.\MessageBreak
3885             This will probably not work, unless the post-\MessageBreak
3886             processing program (dvips, dvipdfm(x), ...) is\MessageBreak
3887             able to create the virtual fonts on the fly}%
3888         \MT@gl@et\MT@warn@tracking@DVI\relax
3889     }
3890     \else
3891     \def\MT@warn@tracking@DVI{%
3892         \ifnum\pdfprotrudechars<\@ne \global\pdfprotrudechars\@ne \fi
3893         \MT@gl@et\MT@warn@tracking@DVI\relax
3894     }
3895     \fi
3896     \ifnum\MT@letterspace=\m@ne
3897     \let\MT@letterspace\MT@letterspace@default
3898     \else
3899     \MT@ls@too@large\MT@letterspace
3900     \fi
3901 }

```

If pdfTeX is too old, we disable tracking, spacing and kerning, and throw an error message.

```

3902 <*package>
3903 {}{
3904     \MT@addto@setup{%
3905         \ifMT@tracking
3906         \MT@error{Tracking only works with pdftex version 1.40\MessageBreak
3907             or newer. Switching it off}{Upgrade pdftex.}%
3908         \else
3909         \MT@info@nl{No tracking (pdftex too old)}
3910         \fi
3911         \ifMT@spacing
3912         \MT@error{Adjustment of interword spacing only works with\MessageBreak
3913             pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3914         \else
3915         \MT@info@nl{No adjustment of interword spacing (pdftex too old)}
3916         \fi
3917         \ifMT@kerning
3918         \MT@error{Character kerning only works with\MessageBreak
3919             pdftex version 1.40 or newer. Switching it off}{Upgrade pdftex.}%
3920         \else
3921         \MT@info@nl{No adjustment of character kerning (pdftex too old)}
3922         \fi
3923     }
3924 }

```

\DisableLigatures is only admissible in the preamble, therefore we can now disable the corresponding macro, if it was never called.

```

3925 \MT@requires@pdftex5{
3926     \MT@addto@setup{%
3927         \ifMT@noligatures \else

```

```

3928     \let\MT@noligatures\relax
3929     \fi
3930   }
3931 } \relax

```

Remove the leading comma in `\MT@active@features`, and set the document switch to true.

```

3932 \MT@addto@setup{%
3933   \ifx\MT@active@features\@empty \else
3934     \edef\MT@active@features{\expandafter\@gobble\MT@active@features}
3935   \fi
3936   \MT@documenttrue
3937 }

```

`\MT@set@babel@context` Interaction with babel.

```

3938 \def\MT@set@babel@context#1{%
3939   \MT@ifdefined@n@TF{MT@babel@#1}{%
3940     \MT@vinfo{*** Changing to language context `#1'\MessageBreak\on@line}%
3941     \expandafter\MT@exp@one@n\expandafter\microtypecontext
3942     \csname MT@babel@#1\endcsname
3943   }{%
3944     \microtypecontext{protrusion=,expansion=,spacing=,kerning=}%
3945   }%
3946 }

```

`\MT@shorthandoff` Active characters can only be switched off if babel isn't loaded after microtype.

```

3947 \@ifpackageloaded{babel}{
3948   \def\MT@shorthandoff#1#2{%
3949     \MT@info@n1{Switching off #1 babel's active characters (#2)}%
3950     \shorthandoff{#2}}
3951 }{
3952   \def\MT@shorthandoff#1#2{%
3953     \MT@error{You must load `babel' before `~\MT@MT'}
3954     {Otherwise, `~\MT@MT' cannot switch off #1 babel's\MessageBreak
3955       active characters.}}
3956 }

```

We patch the language switching commands to enable language-dependent setup.

```

3957 \MT@addto@setup{%
3958   \ifMT@babel
3959     \@ifpackageloaded{babel}{%
3960       \MT@info@n1{Redefining babel's language switching commands}
3961       \let\MT@orig@select@language\select@language
3962       \def\select@language#1{%
3963         \MT@orig@select@language{#1}%
3964         \MT@set@babel@context{#1}%
3965       }
3966       \let\MT@orig@foreign@language\foreign@language
3967       \def\foreign@language#1{%
3968         \MT@orig@foreign@language{#1}%
3969         \MT@set@babel@context{#1}%
3970       }
3971     } \ifMT@kerning

```

Disable French babel's active characters.

```

3972     \MT@if@false
3973     \MT@with@babel@and@T{french} \MT@if@true
3974     \MT@with@babel@and@T{frenchb} \MT@if@true
3975     \MT@with@babel@and@T{francais} \MT@if@true
3976     \MT@with@babel@and@T{canadien} \MT@if@true
3977     \MT@with@babel@and@T{acadian} \MT@if@true
3978     \ifMT@if@~\MT@shorthandoff{French}{:;!?}\fi

```

Disable Turkish babel's active characters.

```
3979 \MT@if@false
3980 \MT@with@babel@and@T{turkish} \MT@if@true
3981 \ifMT@if@MT@shorthandoff{Turkish}{:!=}\fi
3982 \fi
```

In case babel was loaded before microtype:

```
3983 \MT@set@babel@context\language
3984 }{%
3985 \MT@warning@nl{You did not load the babel package.\MessageBreak
3986 The `babel' option won't have any effect}
3987 }
3988 \fi
3989 }
```

Now we close the \fi from \ifMT@draft.

```
3990 \MT@addto@setup{\fi
```

Set up the current font, most likely the normal font. This has to come after all of the setup (including anything from the preamble) has been dealt with.

```
3991 \selectfont}
```

\MT@curr@file This is the current file (hopefully with the correct extension).

```
3992 \edef\MT@curr@file{\jobname.tex}
```

Finally, execute the setup macro at the end of the preamble, and empty it (the combine class calls it repeatedly).

```
3993 </package>
3994 <plain>\MT@requires@latexl{
3995 \AtBeginDocument{\MT@setup@ \MT@glet\MT@setup@ \empty}
3996 <plain>}\relax
```

Warning if \nonfrenchspacing is active, since space factors will be ignored with \pdfadjustinterwordglue>0. Why 1500? Because some packages redefine \frenchspacing.¹³ This has to be checked after the setup has taken place. There still will be a false warning if babel is loaded after microtype (without the babel option).

```
3997 <*package>
3998 \MT@requires@pdftex6{
3999 \AtBeginDocument{%
4000 \ifMT@spacing
4001 \ifMT@babel \else
4002 \ifnum\scode`. > 1500
4003 \MT@ifstreq\MT@sp@context{nonfrench}\relax{%
4004 \MT@warning@nl{%
4005 \string\nonfrenchspacing\space is active. Adjustment of\MessageBreak
4006 interword spacing will disable it. You might want\MessageBreak
4007 to add \@backslashchar\MT@MT context{spacing=nonfrench}'\MessageBreak
4008 to your preamble}%
4009 }%
4010 \fi
4011 \fi
4012 \fi
4013 }
4014 }\relax
4015 </package>
```

13 Cf. the c.t.t. thread ‘\frenchspacing with AMS packages and babel’, started by Philipp Lehman on 16 August 2005, MID: ddtbaj\$rob\$1@online.de

Restore catcodes.

```
4016 \MT@restore@catcodes
```

That was that.

```
4017 </package|letterspace>
```

15 Configuration files

Let's now write the font configuration files.

```
4018 <*config>
```

```
4019
```

15.1 Font sets

We first declare some sets in the main configuration file.

```
4020 <*m-t>
4021 %%% -----
4022 %%% FONT SETS
4023
4024 \DeclareMicrotypeSet{all}
4025 { }
4026
4027 \DeclareMicrotypeSet{allmath}
4028 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1,OML,OMS,U} }
4029
4030 \DeclareMicrotypeSet{alltext}
4031 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
4032
4033 \DeclareMicrotypeSet{bascmath}
4034 { encoding = {OT1,T1,LY1,OT4,QX,T5,OML,OMS},
4035   family   = {rm*,sf*},
4036   series   = {md*},
4037   size     = {normalsize,footnotesize,small,large}
4038 }
4039
4040 \DeclareMicrotypeSet{basictext}
4041 { encoding = {OT1,T1,LY1,OT4,QX,T5},
4042   family   = {rm*,sf*},
4043   series   = {md*},
4044   size     = {normalsize,footnotesize,small,large}
4045 }
4046
4047 \DeclareMicrotypeSet{smallcaps}
4048 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
4049   shape    = {sc}
4050 }
4051
4052 \DeclareMicrotypeSet{footnotesize}
4053 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
4054   size     = {-small}
4055 }
4056
4057 \DeclareMicrotypeSet{scriptsize}
4058 { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1},
```



```

4059     size      = {-footnotesize}
4060   }
4061
4062 \DeclareMicrotypeSet{normalfont}
4063   { font = */*/*/*/* }
4064

```

The default sets.

```

4065 %%% -----
4066 %%% DEFAULT SETS
4067
4068 \DeclareMicrotypeSetDefault[protrusion]{alltext}
4069 \DeclareMicrotypeSetDefault[expansion]{basictext}
4070 \DeclareMicrotypeSetDefault[spacing]{basictext}
4071 \DeclareMicrotypeSetDefault[kerning]{alltext}
4072 \DeclareMicrotypeSetDefault[tracking]{smallcaps}
4073

```

15.2 Font variants and aliases

```

4074 %%% -----
4075 %%% FONT VARIANTS AND ALIASES
4076

```

These are the variants I happen to be using (expert encoding, oldstyle numerals, swashes, alternative, display, inferior and superior numerals):

```

4077 \DeclareMicrotypeVariants{x,j,w,a,d,0,1}
4078

```

Other candidates: 2 (proportional digits), e (engraved), f (Fraktur), g (small text), h (shadow), l (outline), n (informal), p (ornaments), r (roman), s (sans serif), t (typewriter). I've omitted them since they seem hardly be used and/or they are actually more than a variant, i. e., they shouldn't share a file.

Fonts that are 'the same': The Latin Modern fonts, the virtual fonts from the ae and zefonts, and the eco and hfoldsty packages (oldstyle numerals) all inherit the (basic) settings from Computer Modern Roman. Some of them are in part overwritten later.

```

4079 \DeclareMicrotypeAlias{lmr}{cmr} % lmodern
4080 \DeclareMicrotypeAlias{aer}{cmr} % ae
4081 \DeclareMicrotypeAlias{zer}{cmr} % zefonts
4082 \DeclareMicrotypeAlias{cmor}{cmr} % eco
4083 \DeclareMicrotypeAlias{hfor}{cmr} % hfoldsty

```

The packages pxfonts and txfonts fonts inherit Palatino and Times settings respectively, also the T_EX Gyre fonts Pagella and Termes (formerly: qfonts).

```

4084 \DeclareMicrotypeAlias{pxr}{ppl} % pxfonts
4085 \DeclareMicrotypeAlias{qpl}{ppl} % TeX Gyre Pagella (formerly: qfonts/QuasiPalatino)

```

The 'FPL Neu' fonts, a 're-implementation' of Palatino.

```

4086 \DeclareMicrotypeAlias{fp9x}{pplx} % FPL Neu
4087 \DeclareMicrotypeAlias{fp9j}{pplj} % "
4088 \DeclareMicrotypeAlias{txr}{ptm} % txfonts
4089 \DeclareMicrotypeAlias{qtm}{ptm} % TeX Gyre Termes (formerly: qfonts/QuasiTimes)

```

More Times variants, to be checked: pns, mns (TimesNewRomanPS); mnt (TimesNewRomanMT, TimesNRSevenMT), mtm (TimesSmallTextMT); pte (TimesEuropa); ptt (TimesTen); TimesEighteen; TimesModernEF.

The eulervm package virtually extends the Euler fonts.

```

4090 \DeclareMicrotypeAlias{eur}{eur} % Euler VM
4091 \DeclareMicrotypeAlias{zeus}{eus} % "

MicroPress's Charter version (chmath).
4092 \DeclareMicrotypeAlias{chr}{bch} % CH Math

The mathdesign package provides math fonts matching Bitstream Charter and URW
Garamond.
4093 \DeclareMicrotypeAlias{mdbch}{bch} % mathdesign/Charter
4094 \DeclareMicrotypeAlias{mdugm}{ugm} % mathdesign/URW Garamond

URW Letter Gothic is similar enough to Bitstream Letter Gothic to share the config-
uration.
4095 \DeclareMicrotypeAlias{ulg}{blg} % URW LetterGothic -> Bitstream LetterGothic12Pitch

Euro symbol fonts, to save some files.
4096 \DeclareMicrotypeAlias{zpeu}{zpeu} % Adobe Euro sans -> serif
4097 \DeclareMicrotypeAlias{eurosans}{zpeu} % Adobe Euro sans -> serif
4098 \DeclareMicrotypeAlias{euroitcs}{euroitc} % ITC Euro sans -> serif
4099

```

15.3 Interaction with babel

Contexts that are to be set when switching to a language.

```

4100 %%% -----
4101 %%% INTERACTION WITH THE `babel' PACKAGE
4102
4103 \DeclareMicrotypeBabelHook
4104 {english,UKenglish,british,USenglish,american}
4105 {kerning=, spacing=nonfrench}
4106
4107 \DeclareMicrotypeBabelHook
4108 {french,français,acadian,canadien}
4109 {kerning=french, spacing=}
4110
4111 \DeclareMicrotypeBabelHook
4112 {turkish}
4113 {kerning=turkish, spacing=}
4114

```

15.4 Note on admissible characters

All printable ASCII characters are allowed in the settings, with the following exceptions (on the left hand side, the replacements on the right):

```

\ : \textbackslash
{ : \textbraceleft
} : \textbraceright
^ : \textasciicircum
% : \%
# : \#

```

Comma and equal sign must be guarded with braces ('{,}', '{=}') to keep keyval happy.

Character commands are allowed as far as they have been defined in the proper \LaTeX way, that is, when they have been assigned a slot in the font encoding with `\DeclareTextSymbol` or `\DeclareTextComposite`. Characters defined via `\chardef` are also possible.

8-bit characters are also admissible, provided they have been declared in the input encoding file. They should, however, only be used in private configuration files, where the proper input encoding is guaranteed, or else in combination with the ‘inputenc’ key.

```

4131 \DeclareCharacterInheritance
4132 { encoding = T1 }
4133 { A = {\^A,\^A,\^A,\^A,\^A,\r A,\k A,\u A},
4134   a = {\`a,\`a,\`a,\`a,\`a,\r a,\k a,\u a},
4135   C = {\^C,\c C,\v C},
4136   c = {\`c,\c c,\v c},
4137   D = {\v D,\DH},
4138   d = {\v d,\dj},
4139   E = {\^E,\^E,\^E,\^E,\^E,\k E,\v E},
4140   e = {\`e,\`e,\`e,\`e,\`e,\k e,\v e},
4141   f = {027}, % ff
4142   G = {\u G},
4143   g = {\u g},
4144   I = {\^I,\^I,\^I,\^I,\^I,\^I,\^I,\^I},
4145   i = {\`i,\`i,\`i,\`i,\`i,\`i},
4146   j = {\j},
4147   L = {\L,\^L,\v L},
4148   l = {\l,\l,\v l},
4149   N = {\^N,\^N,\v N},

```

```

4150 n = {\`n,\~n,\v n},
4151 O = {\0,\`0,\'0,\^0,\-0,\"0,\H 0},
4152 o = {\o,\`o,\'o,\^o,\-o,\"o,\H o},
4153 R = {\`R,\v R},
4154 r = {\`r,\v r},
4155 S = {\`S,\c S,\v S,\SS},
4156 s = {\`s,\c s,\v s},
4157 T = {\c T,\v T},
4158 t = {\c t,\v t},
4159 U = {\`U,\'U,\^U,\"U,\H U,\r U},
4160 u = {\`u,\'u,\^u,\"u,\H u,\r u},
4161 Y = {\`Y,\"Y},
4162 y = {\`y,\"y},
4163 Z = {\`Z,\.Z,\v Z},
4164 z = {\`z,\.z,\v z}

```

The ‘soft hyphen’ often has reduced right side bearing so that it may already be protruded, hence no inheritance.

```

4165 % - = {127},
4166 }
4167

```

15.5.3 LY1

More characters: 008 (‘fl’), 012 (‘fi’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```

4168 \DeclareCharacterInheritance
4169 { encoding = LY1 }
4170 { A = {\`A,\'A,\^A,\-A,\"A,\r A},
4171   a = {\`a,\'a,\^a,\-a,\"a,\r a},
4172   C = {\c C},
4173   c = {\c c},
4174   D = {\DH},
4175   E = {\`E,\'E,\^E,\"E},
4176   e = {\`e,\'e,\^e,\"e},
4177   f = {011}, % ff
4178   I = {\`I,\'I,\^I,\"I},
4179   i = {\`i,\'i,\^i,\"i,\i},
4180   L = {\L},
4181   l = {\l},
4182   N = {\-N},
4183   n = {\-n},
4184   O = {\`O,\'O,\^O,\-O,\"O,\O},
4185   o = {\`o,\'o,\^o,\-o,\"o,\o},
4186   S = {\v S},
4187   s = {\v s},
4188   U = {\`U,\'U,\^U,\"U},
4189   u = {\`u,\'u,\^u,\"u},
4190   Y = {\`Y,\"Y},
4191   y = {\`y,\"y},
4192   Z = {\v Z},
4193   z = {\v z}
4194 }
4195

```

15.5.4 OT4

The Polish OT1 extension. More interesting characters here: 009 (‘fk’), 012 (‘fi’), 013 (‘fl’), 014 (‘ffi’), 015 (‘ffl’), Æ, æ, Œ, œ.

```

4196 \DeclareCharacterInheritance

```

```

4197 { encoding = OT4 }
4198 { A = {\k A},
4199   a = {\k a},
4200   C = {\'C},
4201   c = {\'c},
4202   E = {\k E},
4203   e = {\k e},
4204   f = {011}, % ff
4205   i = {\i},
4206   j = {\j},
4207   L = {\L},
4208   l = {\l},
4209   N = {\'N},
4210   n = {\'n},
4211   O = {\0,\'0},
4212   o = {\o,\'o},
4213   S = {\'S},
4214   s = {\'s},
4215   Z = {\'Z,\.Z},
4216   z = {\'z,\.z}
4217 }
4218

```

15.5.5 QX

The Central European QX encoding.¹⁴ Ligatures: 009 ('fk'), 012 ('fi'), 013 ('fl'), 014 ('ffi'), 015 ('ffl'), Æ, æ, Œ, œ.

```

4219 \DeclareCharacterInheritance
4220 { encoding = QX }
4221 { A = {\`A,\'A,\^A,\~A,\"A,\k A,\AA},
4222   a = {\`a,\'a,\^a,\~a,\"a,\k a,\aa},
4223   C = {\'C,\c C},
4224   c = {\'c,\c c},
4225   D = {\DH},
4226   E = {\`E,\'E,\^E,\"E,\k E},
4227   e = {\`e,\'e,\^e,\"e,\k e},
4228   f = {011}, % ff
4229   I = {\`I,\'I,\^I,\"I,\k I},
4230   i = {\`i,\'i,\^i,\"i,\k i,\i},
4231   j = {\j},
4232   L = {\L},
4233   l = {\l},
4234   N = {\'N,\~N},
4235   n = {\'n,\~n},
4236   O = {\0,\`0,\'0,\^0,\~0,\"0},
4237   o = {\o,\`o,\'o,\^o,\~o,\"o},

```

The Rumanian `\textcommabelow` accents are actually replacements for the `\c` variants, which had previously (and erroneously¹⁵) been included in QX encoding. They are still kept for backwards compatibility.

```

4238   S = {\'S,\c S,\textcommabelow S,\v S},
4239   s = {\'s,\c s,\textcommabelow s,\v s},
4240   T = {\c T,\textcommabelow T},
4241   t = {\c t,\textcommabelow t},
4242   U = {\`U,\'U,\^U,\"U,\k U},
4243   u = {\`u,\'u,\^u,\"u,\k u},
4244   Y = {\'Y,\"Y},
4245   y = {\'y,\"y},

```

¹⁴ Thanks to *Maciej Eder*.

¹⁵ Cf. <http://tug.org/pipermail/tex-live/2008-August/017204.html>

```

4246 Z = {\'Z,\.Z,\v Z},
4247 z = {\'z,\.z,\v z},
4248 . = \textellipsis
4249 }
4250

```

15.5.6 T5

The Vietnamese encoding T5. It is so crowded with accented and double-accented characters that there is no room for any ligatures.

```

4251 \DeclareCharacterInheritance
4252 { encoding = T5 }
4253 { A = {\^A,\'A,\~A,\h A,\d A,\^A,\u A,
4254       \Acircumflex,\'Acircumflex,\~Acircumflex,\hAcircumflex,\dAcircumflex,
4255       \Abreve,\'Abreve,\~Abreve,\hAbreve,\dAbreve},
4256 a = {\`a,\'a,\~a,\h a,\d a,\^a,\u a,
4257       \acircumflex,\'acircumflex,\~acircumflex,\hacircumflex,\dacircumflex,
4258       \abreve,\'abreve,\~abreve,\h\abreve,\d\abreve},
4259 D = {\DJ},
4260 d = {\dj},
4261 E = {\`E,\'E,\~E,\h E,\d E,\^E,
4262       \Ecircumflex,\'Ecircumflex,\~Ecircumflex,\hEcircumflex,\dEcircumflex},
4263 e = {\`e,\'e,\~e,\h e,\d e,\^e,
4264       \ecircumflex,\'ecircumflex,\~ecircumflex,\hecircumflex,\decircumflex},
4265 I = {\`I,\'I,\~I,\h I,\d I},
4266 i = {\`i,\'i,\~i,\h i,\d i,\i},
4267 O = {\`O,\'O,\~O,\h O,\d O,\^O,\horn O,
4268       \Ocircumflex,\'Ocircumflex,\~Ocircumflex,\hOcircumflex,\dOcircumflex,
4269       \Ohorn,\'Ohorn,\~Ohorn,\hOhorn,\dOhorn},
4270 o = {\`o,\'o,\~o,\h o,\d o,\^o,\horn o,
4271       \ocircumflex,\'ocircumflex,\~ocircumflex,\hocircumflex,\docircumflex,
4272       \ohorn,\'ohorn,\~ohorn,\hohorn,\dohorn},
4273 U = {\`U,\'U,\~U,\h U,\d U,\horn U,
4274       \Uhorn,\'Uhorn,\~Uhorn,\hUhorn,\dUhorn},
4275 u = {\`u,\'u,\~u,\h u,\d u,\horn u,
4276       \uhorn,\'uhorn,\~uhorn,\huhorn,\duhorn},
4277 Y = {\`Y,\'Y,\~Y,\h Y,\d Y},
4278 y = {\`y,\'y,\~y,\h y,\d y}
4279 }
4280
4281 </m-t>

```

15.5.7 Euro symbols

Make Euro symbols settings simpler.

```

4282 < *zpeu >
4283 \DeclareCharacterInheritance
4284 { encoding = U,
4285   family   = {zpeu,zpeus,eurosans} }
4286 { E = 128 }
4287
4288 < /zpeu >
4289 < *mvs >
4290 \DeclareCharacterInheritance
4291 { encoding = OT1,
4292   family   = mvs }
4293 { 164 = {099,100,101} } % \EURhv,\EURcr,\EURtm
4294

```

Since 2006/05/11 (that is, one week after I've added these settings, after the package had been dormant for six years), marvosym's encoding is (correctly) U instead of OT1.

```
4295 \DeclareCharacterInheritance
4296   { encoding = U,
4297     family   = mvs }
4298   { 164 = {099,100,101} }
4299
4300 (/mvs)
```

15.6 Tracking

By default, we only disable the ‘f*’ ligatures, for those fonts that have any. Thus, ligatures and especially kerning for all other characters will be retained.

```
4301 (*m-t)
4302 %%% -----
4303 %%% TRACKING/LETTERSPACING
4304
4305 \SetTracking
4306   [ name      = default,
4307     no ligatures = {f} ]
4308   { encoding   = {OT1,T1,LY1,OT4,QX} }
4309   { }
4310
```

15.7 Font expansion

These are Hàn Thế Thành's original expansion settings. They are used for all fonts (until somebody shows mercy and creates font-specific settings).

```
4311 %%% -----
4312 %%% EXPANSION
4313
4314 \SetExpansion
4315   [ name      = default      ]
4316   { encoding = {OT1,OT4,QX,T1,LY1} }
4317   {
4318     A = 500,      a = 700,
4319     \AE = 500,    \ae = 700,
4320     B = 700,      b = 700,
4321     C = 700,      c = 700,
4322     D = 500,      d = 700,
4323     E = 700,      e = 700,
4324     F = 700,
4325     G = 500,      g = 700,
4326     H = 700,      h = 700,
4327     K = 700,      k = 700,
4328     M = 700,      m = 700,
4329     N = 700,      n = 700,
4330     O = 500,      o = 700,
4331     \OE = 500,    \oe = 700,
4332     P = 700,      p = 700,
4333     Q = 500,      q = 700,
4334     R = 700,
4335     S = 700,      s = 700,
4336     U = 700,      u = 700,
4337     W = 700,      w = 700,
4338     Z = 700,      z = 700,
4339     2 = 700,
```

```

4340     3 = 700,
4341     6 = 700,
4342     8 = 700,
4343     9 = 700
4344   }
4345

```

T5 encoding does not contain \AE, \ae, \OE and \oe.

```

4346 \SetExpansion
4347   [ name      = T5 ]
4348   { encoding = T5 }
4349   {
4350     A = 500,      a = 700,
4351     B = 700,      b = 700,
4352     C = 700,      c = 700,
4353     D = 500,      d = 700,
4354     E = 700,      e = 700,
4355     F = 700,
4356     G = 500,      g = 700,
4357     H = 700,      h = 700,
4358     K = 700,      k = 700,
4359     M = 700,      m = 700,
4360     N = 700,      n = 700,
4361     O = 500,      o = 700,
4362     P = 700,      p = 700,
4363     Q = 500,      q = 700,
4364     R = 700,
4365     S = 700,      s = 700,
4366     U = 700,      u = 700,
4367     W = 700,      w = 700,
4368     Z = 700,      z = 700,
4369     2 = 700,
4370     3 = 700,
4371     6 = 700,
4372     8 = 700,
4373     9 = 700
4374   }
4375
4376 \m-t

```

15.8 Character protrusion

```

4377 %%% -----
4378 %%% PROTRUSION
4379

```

For future historians, Hàn Thế Thành's original settings (from `protcode.tex`, converted to microtype notation).

```

\SetProtrusion
[ name      = thanh ]
{ encoding = OT1 }
{
  A = {50,50},
  F = { ,50},
  J = {50, },
  K = { ,50},
  L = { ,50},
  T = {50,50},
  V = {50,50},
  W = {50,50},
  X = {50,50},

```



```

Y = {50,50},
k = { ,50},
r = { ,50},
t = { ,50},
v = {50,50},
w = {50,50},
x = {50,50},
y = {50,50},
. = { ,700},    {,}= { ,700},
: = { ,500},    ; = { ,500},
! = { ,200},    ? = { ,200},
( = {50, },    ) = { ,50},
- = { ,700},
\textendash    = { ,300},    \textendash    = { ,200},
\textquoteleft = {700, },    \textquoteright = { ,700},
\textquotedblleft = {500, },    \textquotedblright = { ,500}
}

```

15.8.1 Default

The default settings always use the most moderate value.

```

4380 <*cfg-t>
4381 \SetProtrusion
4382 <m-t> [ name = default ]

```

We also create configuration files for the fonts

- Bitstream Charter (NFSS code bch)

```

4383 <bch> [ name = bch-default ]

```

- Bitstream Letter Gothic (blg)

```

4384 <blg> [ name = blg-default ]

```

- Computer Modern Roman (cmr)

```

4385 <cmr> [ name = cmr-default ]

```

- Adobe Garamond (pad, padx, padj)

```

4386 <pad> [ name = pad-default ]

```

- Minion¹⁶ (pmnx, pmnj)

```

4387 <pmn> [ name = pmnj-default ]

```

- Palatino (ppl, pplx, pplj)

```

4388 <ppl> [ name = ppl-default ]

```

- Times (ptm, ptmx, ptmj)

```

4389 <ptm> [ name = ptm-default ]

```

- URW Garamond (ugm)

```

4390 <ugm> [ name = ugm-default ]
4391 <m-t> { encoding = OT1 }
4392 <cmr> { }
4393 <bch|blg|pad|pmn|ugm> { encoding = OT1,
4394 <ppl|ptm> { encoding = {OT1,OT4},

```

16 Contributed by Harald Harders (h.harders@tu-bs.de).

```

4395 <bch>    family = bch }
4396 <blg>    family = blg }
4397 <pad>    family = {pad,padx,padj} }
4398 <pmn>    family = pmnj }
4399 <ppl>    family = {ppl,pplx,pplj} }
4400 <ptm>    family = {ptm,ptmx,ptmj} }
4401 <ugm>    family = ugm }
4402 {
4403 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm>    A = {50,50},
4404 <ugm>    A = {50,100},
4405 <m-t|pad|ptm>    \AE = {50, },
4406 <ugm>    \AE = {150,50},
4407 <ugm>    B = { ,50},
4408 <bch|pad|pmn|ugm>    C = {50, },
4409 <bch|pad|pmn>    D = { ,50},
4410 <ugm>    D = { ,70},
4411 <ugm>    E = { ,50},
4412 <m-t|bch|cmr|pad|pmn|ptm>    F = { ,50},
4413 <ugm>    F = { ,70},
4414 <bch|pad|pmn>    G = {50, },
4415 <ugm>    G = {50,50},
4416 <blg>    I = {150,150},
4417 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    J = {50, },
4418 <bch|blg>    J = {100, },
4419 <!blg>    K = { ,50},
4420 <blg>    K = {50, },
4421 <m-t|bch|cmr|pad|pmn|ppl>    L = { ,50},
4422 <blg>    L = { ,150},
4423 <ptm>    L = { ,80},
4424 <ugm>    L = { ,120},
4425 <bch|pad|pmn|ugm>    O = {50,50},
4426 <pad|pmn>    \OE = {50, },
4427 <ugm>    \OE = {50,50},
4428 <blg>    P = { ,100},
4429 <ugm>    P = { ,50},
4430 <bch|pad|pmn>    Q = {50,70},
4431 <ugm>    Q = {50,50},
4432 <bch>    R = { ,50},
4433 <ugm>    R = { ,70},
4434 <m-t|bch|cmr|pad|pmn|ppl|ptm>    T = {50,50},
4435 <blg>    T = {100,100},
4436 <ugm>    T = {70,70},
4437 <m-t|bch|cmr|pad|pmn|ppl|ptm>    V = {50,50},
4438 <blg|ugm>    V = {70,70},
4439 <m-t|bch|cmr|pad|pmn|ppl|ptm>    W = {50,50},
4440 <ugm>    W = {70,70},
4441 <m-t|bch|cmr|pad|pmn|ppl|ptm>    X = {50,50},
4442 <ugm>    X = {50,70},
4443 <m-t|bch|cmr|pad|pmn|ppl>    Y = {50,50},
4444 <blg|ptm|ugm>    Y = {80,80},
4445 <ugm>    Z = {50,50},
4446 <blg>    f = {150,100},
4447 <blg>    i = {150,150},
4448 <blg>    j = {100,100},
4449 <m-t|bch|cmr|pad|pmn|ppl|ptm>    k = { ,50},
4450 <ugm>    k = { ,70},
4451 <blg>    l = {150,150},
4452 <pmn>    l = { , -50},
4453 <pad|ppl>    p = {50,50},
4454 <ugm>    p = { ,50},
4455 <pad|ppl>    q = {50, },
4456 <!blg>    r = { ,50},
4457 <blg>    r = {100, 80},

```

```

4458 <cmr|pad|pmn>      t = { ,70},
4459 <bch>               t = { ,50},
4460 <blg>               t = {150, 80},
4461 <ugm>               t = { ,100},
4462 <m-t|bch|cmr|pad|pmn|ppl|ptm>    v = {50,50},
4463 <blg>               v = {100,100},
4464 <ugm>               v = {50,70},
4465 <m-t|bch|cmr|pad|pmn|ppl|ptm>    w = {50,50},
4466 <ugm>               w = {50,70},
4467 <!blg>              x = {50,50},
4468 <blg>               x = {100,100},
4469 <m-t|bch|pad|pmn>    y = { ,50},
4470 <blg>               y = { 50,100},
4471 <cmr|ppl|ptm>        y = {50,70},
4472 <ugm>               y = { ,70},

4473 <cmr>               0 = { ,50},
4474 <m-t>               1 = {50,50},
4475 <bch|blg|pad|ptm|ugm>      1 = {150,150},
4476 <cmr>               1 = {100,200},
4477 <pmn>               1 = { ,50},
4478 <ppl>               1 = {100,100},
4479 <bch|cmr|pad|ugm>       2 = {50,50},
4480 <blg>               2 = { ,100},
4481 <bch|pmn>           3 = {50, },
4482 <cmr|pad|ugm>        3 = {50,50},
4483 <blg>               3 = {100, },
4484 <m-t|pad>           4 = {50,50},
4485 <bch>               4 = {100,50},
4486 <blg>               4 = {100, },
4487 <cmr|ugm>           4 = {70,70},
4488 <pmn>               4 = {50, },
4489 <ptm>               4 = {70, },
4490 <cmr>               5 = { ,50},
4491 <pad>               5 = {50,50},
4492 <bch>               6 = {50, },
4493 <cmr>               6 = { ,50},
4494 <pad>               6 = {50,50},
4495 <m-t>               7 = {50,50},
4496 <bch|pad|pmn|ugm>      7 = {50,80},
4497 <blg>               7 = {100,100},
4498 <cmr|ptm>           7 = {50,100},
4499 <ppl>               7 = { ,50},
4500 <cmr>               8 = { ,50},
4501 <bch|pad>           9 = {50,50},
4502 <cmr>               9 = { ,50},
4503 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    . = { ,700},
4504 <bch>               . = { ,600},
4505 <blg>               . = {400,500},
4506 <!blg>             {,}= { ,500},
4507 <blg>               {,}= {300,400},
4508 <m-t|cmr|pad|pmn|ppl|ptm|ugm>    : = { ,500},
4509 <bch>               : = { ,400},
4510 <blg>               : = {300,400},
4511 <m-t|bch|pad|pmn|ptm>    ; = { ,300},
4512 <blg>               ; = {200,300},
4513 <cmr|ppl>           ; = { ,500},
4514 <ugm>               ; = { ,400},
4515 <!blg>              != { ,100},
4516 <blg>               != {200,200},
4517 <m-t|pad|pmn|ptm>      ? = { ,100},
4518 <bch|cmr|ppl|ugm>      ? = { ,200},
4519 <blg>               ? = {150,150},

```

```

4520 <pmn>      " = {300,300},
4521 <m-t|bch|cmr|pad|pmn|ppl>    @ = {50,50},
4522 <ptm>      @ = {100,100},
4523 <m-t|bch|blg|cmr|pad|pmn|ppl|ptm> ~ = {200,250},
4524 <ugm>      ~ = {300,350},
4525 <pad|ppl|ptm>    & = {50,100},
4526 <ugm>      & = { ,100},
4527 <m-t|cmr|pad|pmn>    \% = {50,50},
4528 <bch>      \% = { ,50},
4529 <ppl|ptm>    \% = {100,100},
4530 <ugm>      \% = {50,100},
4531 <blg>      \# = {100,100},
4532 <m-t|ppl|ptm|ugm>    * = {200,200},
4533 <bch|pmn>    * = {200,300},
4534 <blg>      * = {150,200},
4535 <cmr|pad>    * = {300,300},
4536 <m-t|cmr|ppl|ptm>    + = {250,250},
4537 <bch>      + = {150,250},
4538 <pad>      + = {300,300},
4539 <blg|pmn>    + = {150,200},
4540 <ugm>      + = {250,300},
4541 <blg|ugm>    {=} = {200,200},
4542 <m-t|pad|pmn|ptm>    ( = {100,  },  ) = { ,200},
4543 <bch|ugm>    ( = {200,  },  ) = { ,200},
4544 <cmr|blg>    ( = {300,  },  ) = { ,300},
4545 <ppl>    ( = {100,  },  ) = { ,300},
4546 <bch|pmn>    [ = {100,  },  ] = { ,100},
4547 <blg>    [ = {300,100},  ] = { ,300},

4548 <m-t|pad|pmn|ptm>    / = {100,200},
4549 <bch>      / = { ,200},
4550 <blg>      / = {300,300},
4551 <cmr|ppl>    / = {200,300},
4552 <ugm>      / = {100,300},
4553 <m-t|ptm>    - = {500,500},
4554 <bch|cmr|ppl>    - = {400,500},
4555 <blg>      - = {300,400},
4556 <pad>      - = {300,500},
4557 <pmn>      - = {200,400},
4558 <ugm>      - = {500,600},
4559 <blg>      < = {200,100},    > = {100,200},
4560 <blg>      _ = {150,250},
4561 <blg>      | = {250,250},
4562 <m-t|pmn>    \textendash      = {200,200},    \textendash      = {150,150},
4563 <bch>    \textendash      = {200,300},    \textendash      = {150,250},
4564 <cmr>    \textendash      = {400,300},    \textendash      = {300,200},
4565 <pad|ppl|ptm>    \textendash      = {300,300},    \textendash      = {200,200},
4566 <ugm>    \textendash      = {250,300},    \textendash      = {250,250},

```

Why settings for left *and* right quotes? Because in some languages they might be used like that (see the csquotes package for examples).

```

4567 <m-t|bch|pmn>    \textquoteleft = {300,400},    \textquoteright = {300,400},
4568 <blg>    \textquoteleft = {400,600},    \textquoteright = {400,600},
4569 <cmr>    \textquoteleft = {500,700},    \textquoteright = {500,600},
4570 <pad|ppl>    \textquoteleft = {500,700},    \textquoteright = {500,700},
4571 <ptm>    \textquoteleft = {500,500},    \textquoteright = {300,500},
4572 <ugm>    \textquoteleft = {300,600},    \textquoteright = {300,600},
4573 <m-t|bch|pmn>    \textquotedblleft = {300,300},    \textquotedblright = {300,300}
4574 <blg>    \textquotedblright = {300,400}
4575 <cmr>    \textquotedblleft = {500,300},    \textquotedblright = {200,600}
4576 <pad|ppl|ptm>    \textquotedblleft = {300,400},    \textquotedblright = {300,400}
4577 <ugm>    \textquotedblleft = {400,400},    \textquotedblright = {400,400}

```

```
4578 }
4579
```

Greek uppercase letters are in OT1 encoding only.

```
4580 (*cmr)
4581 \SetProtrusion
4582 [ name      = cmr-OT1,
4583   load      = cmr-default ]
4584 { encoding = {OT1,OT4},
4585   family    = cmr }
4586 {
4587   \AE = { 50, },
4588   "00 = { ,150}, % \Gamma
4589   "01 = {100,100}, % \Delta
4590   "02 = { 50, 50}, % \Theta
4591   "03 = {100,100}, % \Lambda
4592   "06 = { 50, 50}, % \Sigma
4593   "07 = {100,100}, % \Upsilon
4594   "08 = { 50, 50}, % \Phi
4595   "09 = { 50, 50} % \Psi
```

Remaining slots can be found in the source file.

```
4596 }
4597
4598 (/cmr)
```

T1 and LY1 encodings contain some more characters. The default list will be loaded first.

```
4599 \SetProtrusion
4600 <m-t> [ name      = T1-default,
4601 <bch> [ name      = bch-T1,
4602 <blg> [ name      = blg-T1,
4603 <cmr> [ name      = cmr-T1,
4604 <pad> [ name      = pad-T1,
4605 <pmn> [ name      = pmnj-T1,
4606 <ppl> [ name      = ppl-T1,
4607 <ptm> [ name      = ptm-T1,
4608 <ugm> [ name      = ugm-T1,
4609 <m-t>   load      = default ]
4610 <bch>   load      = bch-default ]
4611 <blg>   load      = blg-default ]
4612 <cmr>   load      = cmr-default ]
4613 <pad>   load      = pad-default ]
4614 <pmn>   load      = pmnj-default ]
4615 <ppl>   load      = ppl-default ]
4616 <ptm>   load      = ptm-default ]
4617 <ugm>   load      = ugm-default ]
4618 <m-t>   { encoding = {T1,LY1} }
4619 <bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
4620 <blg|ptm|ugm> { encoding = {T1},
4621 <bch>   family    = bch }
4622 <blg>   family    = blg }
4623 <cmr>   family    = cmr }
4624 <pad>   family    = {pad,padx,padj} }
4625 <pmn>   family    = pmnj }
4626 <ppl>   family    = {ppl,pplx,pplj} }
4627 <ptm>   family    = {ptm,ptmx,ptmj} }
4628 <ugm>   family    = ugm }
4629 {
4630 <cmr>   \AE = {50, },
4631 <bch>   \OE = {50, },
4632 <pmn>   \TH = { ,50},
```

```

4633 <blg> \v L = { ,250},
4634 <blg> \v d = { ,250},
4635 <blg> \v l = { ,250},
4636 <blg> \v t = { ,250},
4637 <blg> 127 = {300,400},
4638 <blg> 156 = {100, }, % IJ
4639 <blg> 188 = { 80, 80}, % ij
4640 <m-t|bch|pad|pmn|ppl|ptm> _ = {100,100},
4641 <cmr> _ = {200,200},
4642 <ugm> _ = {100,200},
4643 <m-t|pad|pmn|ptm> \textbackslash = {100,200},
4644 <bch> \textbackslash = {150,200},
4645 <blg> \textbackslash = {250,300},
4646 <cmr|ppl> \textbackslash = {200,300},
4647 <ugm> \textbackslash = {100,300},
4648 <ugm> \textbar = {200,200},
4649 <blg> \textendash = {300,300}, \textemdash = {150,150},
4650 <blg> \textquotedbl = {300,400}, \textquotedblleft = {300,400},
4651 <cmr> \textquotedbl = {300,300}, \textquotedblleft = {200,600},

```

The EC fonts do something weird: they insert an implicit kern between quote and boundary character. Therefore, we must override the settings from OT1.

```

4652 <m-t|cmr|pad|ppl|ptm|ugm> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4653 <blg> \quotesinglbase = {400,400}, \quotedblbase = {300,400},
4654 <bch|pmn> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4655 <m-t|bch|pmn> \guilsinglleft = {400,300}, \guilsingright = {300,400},
4656 <blg> \guilsinglleft = {300,500}, \guilsingright = {300,500},
4657 <cmr|pad|ppl|ptm> \guilsinglleft = {400,400}, \guilsingright = {300,500},
4658 <ugm> \guilsinglleft = {400,400}, \guilsingright = {300,600},
4659 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4660 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4661 <bch|pmn> \guillemotleft = {200,200}, \guillemotright = {150,300},
4662 <blg|pad|ppl|ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4663 <ugm> \guillemotleft = {300,400}, \guillemotright = {300,400},
4664 <m-t|bch|cmr|pad|pmn|ppl|ugm> \textexclamdown = {100, }, \textquestiondown = {100, },
4665 <blg> \textexclamdown = {200, }, \textquestiondown = {100, },
4666 <ptm> \textexclamdown = {200, }, \textquestiondown = {200, },
4667 <m-t|cmr|pad|ppl|ptm|ugm> \textbraceleft = {400,200}, \textbraceright = {200,400},
4668 <bch|blg|pmn> \textbraceleft = {200, }, \textbraceright = { ,300},
4669 <m-t|bch|cmr|pad|ppl|ptm|ugm> \textless = {200,100}, \textgreater = {100,200}
4670 <pmn> \textless = {100, }, \textgreater = { ,100},
4671 <pmn> \textvisiblespace = {100,100} % not in LY1
4672 }
4673

```

The lmodern fonts used to restore the original settings from OT1 fonts. Now, they require even other settings, though.

```

4674 <(*cmr>
4675 \SetProtrusion
4676 [ name = lmr-T1,
4677   load = cmr-T1 ]
4678 { encoding = {T1,LY1},
4679   family = lmr }
4680 {
4681   \textquotedblleft = {300,400}, \textquotedblright = {300,400}
4682 }
4683
4684 </cmr>

```

Settings for the QX encoding (generic and Times). It also includes some glyphs otherwise in TS1.

```

4685 <*m-t|ptm>
4686 \SetProtrusion
4687 <m-t> [ name = QX-default,
4688 <ptm> [ name = ptm-QX,
4689 <m-t> load = default ]
4690 <ptm> load = ptm-default ]
4691 <m-t> { encoding = QX }
4692 <ptm> { encoding = QX,
4693 <ptm> family = {ptm,ptmx,ptmj} }
4694 {
4695 <ptm> * = {200,200},
4696 {=} = {100,100},
4697 \textunderscore = {100,100},
4698 \textbackslash = {100,200},
4699 \quotedblbase = {400,400},
4700 <m-t> \guillemotleft = {200,200}, \guillemotright = {200,200},
4701 <ptm> \guillemotleft = {300,300}, \guillemotright = {200,400},
4702 \textexclamdown = {100, }, \textquestiondown = {100, },
4703 <m-t> \textbraceleft = {400,200}, \textbraceright = {200,400},
4704 <ptm> \textbraceleft = {200,200}, \textbraceright = {200,300},
4705 \textless = {200,100}, \textgreater = {100,200},
4706 \textminus = {200,200}, \textdegree = {300,300},
4707 <m-t> \copyright = {100,100}, \textregistered = {100,100}
4708 <ptm> \copyright = {100,150}, \textregistered = {100,150},
4709 <ptm> \textxgeq = { ,100}, \textxleq = {100, },
4710 <ptm> \textalpha = { , 50}, \textDelta = { 70, 70},
4711 <ptm> \textpi = { 50, 80}, \textSigma = { , 70},
4712 <ptm> \textmu = { , 80}, \texteuro = { 50, 50},
4713 <ptm> \textellipsis = {150,200}, \textasciitilde = { 80, 80},
4714 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
4715 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
4716 <ptm> \textdiv = { 50,150}, \textsection = { 80, 80},
4717 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
4718 <ptm> \textbullet = {150,150}, \textperiodcentered = {300,300},
4719 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
4720 <ptm> \textperthousand = { ,50}
4721 }
4722
4723 </m-t|ptm>

```

T5 is based on OT1; it shares some but not all extra characters of T1. All accented characters are already taken care of by the inheritance list.

```

4724 <*cmr|bch>
4725 \SetProtrusion
4726 <cmr> [ name = cmr-T5,
4727 <cmr> load = cmr-default ]
4728 <bch> [ name = bch-T5,
4729 <bch> load = bch-default ]
4730 { encoding = T5,
4731 <cmr> family = cmr }
4732 <bch> family = bch }
4733 {
4734 <bch> _ = {100,100},
4735 <bch> \textbackslash = {150,200},
4736 <cmr> \textbackslash = {200,300},
4737 <cmr> \textquotedblleft = {200,600},
4738 <cmr> \textquotedbl = {300,300},
4739 <bch> \quotesinglbase = {400,400}, \quotedblbase = {300,300},
4740 <cmr> \quotesinglbase = {400,400}, \quotedblbase = {400,400},
4741 <bch> \guilsinglleft = {400,300}, \guilsinglright = {300,400},
4742 <cmr> \guilsinglleft = {400,400}, \guilsinglright = {300,500},
4743 <bch> \guillemotleft = {200,200}, \guillemotright = {150,300},

```

```

4744 <cmr> \guillemotleft = {300,200}, \guillemotright = {100,400},
4745 <bch> \textbraceleft = {200, }, \textbraceright = { ,300},
4746 <cmr> \textbraceleft = {400,200}, \textbraceright = {200,400},
4747 \textless = {200,100}, \textgreater = {100,200}
4748 }
4749
4750 </cmr|bch>
4751 <*pmn>
4752 \SetProtrusion
4753 [ name = pmnx-OT1,
4754 load = pmnj-default ]
4755 { encoding = OT1,
4756 family = pmnx }
4757 {
4758 1 = {230,180}
4759 }
4760
4761 \SetProtrusion
4762 [ name = pmnx-T1,
4763 load = pmnj-T1 ]
4764 { encoding = {T1,LY1},
4765 family = pmnx }
4766 {
4767 1 = {230,180}
4768 }
4769
4770 </pmn>

```

Times is the default font for LY1, therefore we provide settings for the additional characters in this encoding, too.

```

4771 <*ptm>
4772 \SetProtrusion
4773 [ name = ptm-LY1,
4774 load = ptm-T1 ]
4775 { encoding = LY1,
4776 family = {ptm,ptmx,ptmj} }
4777 {
4778 - = {100,100},
4779 \texttrademark = {100,100},
4780 \textregistered = {100,100},
4781 \textcopyright = {100,100},
4782 \textdegree = {300,300},
4783 \textminus = {200,200},
4784 \textellipsis = {150,200},
4785 % \texteuro = { , }, % ?
4786 \textcent = {100,100},
4787 \textquotesingle = {500,500},
4788 \textflorin = { 50, 70},
4789 \textdagger = {150,150},
4790 \textdaggerdbl = {100,100},
4791 \textperthousand = { , 50},
4792 \textbullet = {150,150},
4793 \textonesuperior = {100,100},
4794 \texttwosuperior = { 50, 50},
4795 \textthreesuperior = { 50, 50},
4796 \textperiodcentered = {300,300},
4797 \textplusminus = { 50, 80},
4798 \textmultiply = {100,100},
4799 \textdivide = { 50,150}

```

Remaining slots in the source file.

```

4800 }

```



```
4801
4802 </ptm>
```

15.8.2 Italics

To find default settings for italic is difficult, since the character shapes and their behaviour at the beginning or end of line may be wildly different for different fonts. Therefore, we leave the letters away, and only set up the punctuation characters.

```
4803 \SetProtrusion
4804 <m-t> [ name      = OT1-it   ]
4805 <bch> [ name      = bch-it   ]
4806 <blg> [ name      = blg-it,
4807       load      = blg-default ]
4808 <cmr> [ name      = cmr-it   ]
4809 <pad> [ name      = pad-it   ]
4810 <pmn> [ name      = pmnj-it  ]
4811 <ppl> [ name      = ppl-it   ]
4812 <ptm> [ name      = ptm-it   ]
4813 <ugm> [ name      = ugm-it   ]
4814 <m-t|bch|blg|pad|pmn|ugm> { encoding = OT1,
4815 <ppl|ptm> { encoding = {OT1,OT4},
4816 <bch>      family   = bch,
4817 <blg>      family   = blg,
4818 <pad>      family   = {pad,padx,padj},
4819 <pmn>      family   = pmnj,
4820 <ppl>      family   = {ppl,pplx,pplj},
4821 <ptm>      family   = {ptm,ptmx,ptmj},
4822 <ugm>      family   = ugm,
4823 <m-t|bch|pad|pmn|ppl|ptm> shape   = {it,sl} }
4824 <blg|ugm> shape   = it }
4825 <cmr> { }
4826 {
4827 <cmr|ptm> A = {100,50},
4828 <pad|pmn> A = {50, },
4829 <ugm>     A = { ,150},
4830 <ppl>     A = {50,50},
4831 <ptm>     \AE = {100, },
4832 <pad|ppl> \AE = {50, },
4833 <pmn>     \AE = { , -50},
4834 <cmr|pad|ppl|ptm> B = {50, },
4835 <pmn>     B = {20,-50},
4836 <bch|ppl|ptm|ugm> C = {50, },
4837 <cmr|pad> C = {100, },
4838 <pmn>     C = {50,-50},
4839 <cmr|pad|ppl|ptm> D = {50,50},
4840 <pmn>     D = {20, },
4841 <cmr|pad|ppl|ptm> E = {50, },
4842 <pmn>     E = {20,-50},
4843 <cmr|pad|ptm> F = {100, },
4844 <pmn>     F = {10, },
4845 <ppl>     F = {50, },
4846 <bch|ppl|ptm|ugm> G = {50, },
4847 <cmr|pad> G = {100, },
4848 <pmn>     G = {50,-50},
4849 <cmr|pad|ppl|ptm> H = {50, },
4850 <cmr|pad|ptm> I = {50, },
4851 <pmn>     I = {20,-50},
4852 <cmr|ptm> J = {100, },
4853 <pad>     J = {50, },
4854 <pmn>     J = {20, },
4855 <cmr|pad|ppl|ptm> K = {50, },
```

```

4856 <pmn>      K = {20,  },
4857 <cmr|pad|ppl|ptm>  L = {50,  },
4858 <pmn>      L = {20,50},
4859 <ugm>      L = {  ,100},
4860 <cmr|ptm>   M = {50,  },
4861 <pmn>      M = {  ,-30},
4862 <cmr|ptm>   N = {50,  },
4863 <pmn>      N = {  ,-30},
4864 <bch|pmn|ppl|ptm>  O = {50,  },
4865 <cmr|pad>   O = {100,  },
4866 <ugm>      O = {70,50},
4867 <pmn|ppl|ptm> \OE = {50,  },
4868 <pad>      \OE = {100,  },
4869 <cmr|pad|ppl|ptm>  P = {50,  },
4870 <pmn>      P = {20,-50},
4871 <bch|pmn|ppl|ptm>  Q = {50,  },
4872 <cmr|pad>   Q = {100,  },
4873 <ugm>      Q = {70,50},
4874 <cmr|pad|ppl|ptm>  R = {50,  },
4875 <pmn>      R = {20,  },
4876 <bch|cmr|pad|ppl|ptm> S = {50,  },
4877 <pmn>      S = {20,-30},
4878 <bch|cmr|pad|ppl|ptm> $ = {50,  },
4879 <pmn>      $ = {20,-30},
4880 <bch|pmn|ugm>     T = {70,  },
4881 <cmr|pad|ppl|ptm>  T = {100,  },
4882 <cmr|pad|ppl|ptm>  U = {50,  },
4883 <pmn>      U = {50,-50},
4884 <cmr|pad|pmn|ugm>  V = {100,  },
4885 <ppl|ptm>      V = {100,50},
4886 <cmr|pad|pmn|ugm>  W = {100,  },
4887 <ppl>        W = {50,  },
4888 <ptm>        W = {100,50},
4889 <cmr|ppl|ptm>     X = {50,  },
4890 <cmr|ptm>      Y = {100,  },
4891 <pmn>      Y = {50,  },
4892 <ppl>      Y = {100,50},
4893 <pmn>      Z = {  ,-50},
4894 <pmn>      d = {  ,-50},
4895 <pad|pmn>     f = {  ,-100},
4896 <pmn>      i = {  ,-30},
4897 <pmn>      j = {  ,-30},
4898 <pmn>      l = {  ,-100},
4899 <bch>      o = {50,50},
4900 <bch>      p = {  ,50},
4901 <pmn>      p = {-50,  },
4902 <bch>      q = {50,  },
4903 <pmn>      r = {  ,50},
4904 <bch>      t = {  ,50},
4905 <pmn|ugm>    v = {50,  },
4906 <bch>      w = {  ,50},
4907 <pmn|ugm>    w = {50,  },
4908 <bch>      y = {  ,50},
4909 <cmr>      O = {100,  },
4910 <bch|ptm>    1 = {150,100},
4911 <cmr>      1 = {200,50},
4912 <pad>      1 = {150,  },
4913 <pmn>      1 = {50,  },
4914 <ppl>      1 = {100,  },
4915 <ugm>      1 = {150,150},
4916 <cmr>      2 = {100,-100},
4917 <pad|ppl|ptm>    2 = {50,  },
4918 <pmn>      2 = {-50,  },

```

```

4919 <bch>      3 = {50, },
4920 <cmr>      3 = {100,-100},
4921 <pmn>      3 = {-100, },
4922 <ptm>      3 = {100,50},
4923 <bch>      4 = {100, },
4924 <cmr|pad>  4 = {150, },
4925 <ppl|ptm>  4 = {50, },
4926 <cmr>      5 = {100, },
4927 <ptm>      5 = {50, },
4928 <bch>      6 = {50, },
4929 <cmr>      6 = {100, },
4930 <bch|pad|ptm> 7 = {100, },
4931 <cmr>      7 = {200,-150},
4932 <pmn>      7 = {20, },
4933 <ppl>      7 = {50, },
4934 <cmr>      8 = {50,-50},
4935 <cmr>      9 = {100,-100},
4936 <m-t|cmr|pad|pmn|ppl> . = { ,500},
4937 <blg>      . = {400,600},
4938 <bch|ptm|ugm> . = { ,700},
4939 <blg>      {,}= {300,500},
4940 <m-t|cmr|pad|pmn|ppl> {,}= { ,500},
4941 <bch|ugm>   {,}= { ,600},
4942 <ptm>      {,}= { ,700},
4943 <m-t|cmr|pad|ppl>   : = { ,300},
4944 <bch|ugm>   : = { ,400},
4945 <pmn>      : = { ,200},
4946 <ptm>      : = { ,500},
4947 <m-t|cmr|pad|ppl>   ; = { ,300},
4948 <bch|ugm>   ; = { ,400},
4949 <pmn>      ; = { ,200},
4950 <ptm>      ; = { ,500},
4951 <ptm>      ! = { ,100},
4952 <bch>      ? = { ,200},
4953 <ptm>      ? = { ,100},
4954 <ppl>      ? = { ,300},
4955 <pmn>      " = {400,200},
4956 <m-t|pad|pmn|ppl|ptm> & = {50,50},
4957 <bch>      & = { ,80},
4958 <cmr>      & = {100,50},
4959 <ugm>      & = {50,100},
4960 <m-t|cmr|pad|pmn>   \% = {100, },
4961 <bch>      \% = {50,50},
4962 <ppl|ptm>   \% = {100,100},
4963 <ugm>      \% = {100,50},
4964 <m-t|pmn|ppl>      * = {200,200},
4965 <bch>      * = {300,200},
4966 <cmr>      * = {400,100},
4967 <pad>      * = {500,100},
4968 <ptm|ugm>   * = {400,200},
4969 <m-t|cmr|pmn|ppl>   + = {150,200},
4970 <bch|ugm>   + = {250,250},
4971 <pad|ptm>   + = {250,200},
4972 <m-t|pad|pmn|ppl>   @ = {50,50},
4973 <bch>      @ = {80,50},
4974 <cmr>      @ = {200,50},
4975 <ptm>      @ = {150,150},
4976 <m-t|bch|ugm>   ~ = {150,150},
4977 <cmr|pad|pmn|ppl|ptm> ~ = {200,150},
4978 <ugm>      {=}= {200,200},
4979 <!blg>      ( = {200, }, ) = { ,200},
4980 <m-t|cmr|pad|ppl|ptm|ugm> / = {100,200},
4981 <bch>      / = { ,150},

```

```

4982 <pmn>      / = {100,150},
4983 <m-t>       - = {300,300},
4984 <bch|pad>    - = {300,400},
4985 <pmn>       - = {200,300},
4986 <cmr>       - = {500,300},
4987 <ppl>       - = {300,500},
4988 <ptm>       - = {500,500},
4989 <ugm>       - = {400,700},
4990 <blg>      - = {0,300},
4991 <m-t|pmn>    \textendash      = {200,200},    \textemdash      = {150,150},
4992 <bch>        \textendash      = {200,300},    \textemdash      = {150,200},
4993 <cmr>        \textendash      = {500,300},    \textemdash      = {400,200},
4994 <pad|ppl|ptm|ugm> \textendash      = {300,300},    \textemdash      = {200,200},
4995 <m-t|bch|pmn|ugm> \textquoteleft = {400,200},    \textquoteright = {400,200},
4996 <blg>        \textquoteleft = {400,400},    \textquoteright = {400,400},
4997 <cmr|pad>    \textquoteleft = {800,200},    \textquoteright = {800,200},
4998 <ppl>        \textquoteleft = {700,400},    \textquoteright = {700,400},
4999 <ptm>        \textquoteleft = {800,500},    \textquoteright = {800,500},
5000 <m-t|bch|pmn> \textquotedblleft = {400,200}, \textquotedblright = {400,200}
5001 <blg>        \textquotedblright = {300,300}
5002 <cmr>        \textquotedblleft = {700,100},    \textquotedblright = {500,300}
5003 <pad>        \textquotedblleft = {700,200},    \textquotedblright = {700,200}
5004 <ppl>        \textquotedblleft = {500,300},    \textquotedblright = {500,300}
5005 <ptm>        \textquotedblleft = {700,400},    \textquotedblright = {700,400}
5006 <ugm>        \textquotedblleft = {600,200},    \textquotedblright = {600,200}
5007     }
5008
5009 <*cmr>
5010 \SetProtrusion
5011     [ name      = cmr-it-OT1,
5012       load      = cmr-it  ]
5013     { encoding = {OT1,OT4},
5014       family   = cmr,
5015       shape    = it      }
5016     {
5017       \AE = {100,  },
5018       \OE = {100,  },
5019       "00 = {200,150}, % \Gamma
5020       "01 = {150,100}, % \Delta
5021       "02 = {150, 50}, % \Theta
5022       "03 = {150, 50}, % \Lambda
5023       "04 = {100,100}, % \Xi
5024       "05 = {100,100}, % \Pi
5025       "06 = {100, 50}, % \Sigma
5026       "07 = {200,150}, % \Upsilon
5027       "08 = {150, 50}, % \Phi
5028       "09 = {150,100}, % \Psi
5029       "0A = { 50, 50} % \Omega
5030     }
5031
5032 </cmr>
5033 \SetProtrusion
5034 <m-t> [ name      = T1-it-default,
5035 <bch> [ name      = bch-it-T1,
5036 <blg> [ name      = blg-it-T1,
5037 <cmr> [ name      = cmr-it-T1,
5038 <pad> [ name      = pad-it-T1,
5039 <pmn> [ name      = pmn-j-it-T1,
5040 <ppl> [ name      = ppl-it-T1,
5041 <ptm> [ name      = ptm-it-T1,
5042 <ugm> [ name      = ugm-it-T1,
5043 <m-t> [ load      = OT1-it  ]
5044 <bch> [ load      = bch-it  ]

```

```

5045 <blg>      load      = blg-T1   ]
5046 <cmr>      load      = cmr-it   ]
5047 <pmn>      load      = pmnj-it  ]
5048 <pad>      load      = pad-it   ]
5049 <ppl>      load      = ppl-it   ]
5050 <ptm>      load      = ptm-it   ]
5051 <ugm>      load      = ugm-it   ]
5052 <m-t|bch|cmr|pad|pmn|ppl> { encoding = {T1,LY1},
5053 <blg|ptm|ugm> { encoding = T1,
5054 <bch>      family    = bch,
5055 <blg>      family    = blg,
5056 <cmr>      family    = cmr,
5057 <pmn>      family    = pmnj,
5058 <pad>      family    = {pad,padx,padj},
5059 <ppl>      family    = {ppl,pplx,pplj},
5060 <ptm>      family    = {ptm,ptmx,ptmj},
5061 <ugm>      family    = ugm,
5062 <m-t|bch|pad|pmn|ppl|ptm> shape    = {it,sl}  }
5063 <blg|cmr|ugm> shape    = it      }
5064 {
5065 <m-t|bch|pmn>      _ = { ,100},
5066 <blg>              _ = {0,300},
5067 <cmr|ugm>          _ = {100,200},
5068 <pad|ppl|ptm>      _ = {100,100},
5069 <blg>              . = {400,600},
5070 <blg>              {,}= {300,500},
5071 <cmr>              \AE = {100,  },
5072 <bch>              \OE = { 50,  },
5073 <cmr>              \OE = {100,  },
5074 <pmn>              031 = { , -100}, % ffl
5075 <cmr|ptm>          156 = {100, }, % IJ
5076 <pad>              156 = {50,  }, % IJ
5077 <pmn>              156 = {20,  }, % IJ
5078 <pmn>              188 = { , -30}, % ij
5079 <pmn>              \v t = { ,100},
5080 <m-t|pad|ppl|ptm> \textbackslash = {100,200},
5081 <cmr|ugm>          \textbackslash = {300,300},
5082 <bch>              \textbackslash = {150,150},
5083 <pmn>              \textbackslash = {100,150},
5084 <ugm>              \textbar      = {200,200},
5085 <cmr>              \textquotedblleft = {500,300},
5086 <blg>              \textquoteleft  = {400,400}, \textquoteright = {400,400},
5087 <blg>              \textquotedbl  = {300,300}, \textquotedblleft = {300,300},
5088 <blg>              \textquotedblright = {300,300}, \quotedblbase = {200,600},
5089 <m-t|ptm>          \quotesinglbase = {300,700}, \quotedblbase = {400,500},
5090 <cmr>              \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5091 <bch|pmn>          \quotesinglbase = {200,500}, \quotedblbase = {150,500},
5092 <pad|ppl>          \quotesinglbase = {500,500}, \quotedblbase = {400,400},
5093 <ugm>              \quotesinglbase = {300,700}, \quotedblbase = {300,500},
5094 <m-t|ppl|ptm>      \guilsinglleft = {400,400}, \guilsinglright = {300,500},
5095 <bch|pmn>          \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5096 <cmr>              \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5097 <pad>              \guilsinglleft = {500,400}, \guilsinglright = {300,500},
5098 <ugm>              \guilsinglleft = {400,400}, \guilsinglright = {300,600},
5099 <m-t|ppl>          \guillemotleft = {300,300}, \guillemotright = {300,300},
5100 <bch|pmn>          \guillemotleft = {200,300}, \guillemotright = {150,400},
5101 <cmr>              \guillemotleft = {400,100}, \guillemotright = {200,300},
5102 <pad>              \guillemotleft = {300,300}, \guillemotright = {200,400},
5103 <ptm>              \guillemotleft = {300,400}, \guillemotright = {200,400},
5104 <ugm>              \guillemotleft = {300,400}, \guillemotright = {300,400},
5105 <m-t|pad|ppl|ugm> \textexclamdown = {100,  }, \textquestiondown = {200,  },
5106 <cmr|ptm>          \textexclamdown = {200,  }, \textquestiondown = {200,  },
5107 <pmn>              \textexclamdown = {-50,  }, \textquestiondown = {-50,  },

```

```

5108 <m-t|ppl|ugm> \textbraceleft = {200,100}, \textbraceright = {200,200},
5109 <bch|pmn> \textbraceleft = {200, }, \textbraceright = { ,200},
5110 <cmr|pad|ptm> \textbraceleft = {400,100}, \textbraceright = {200,200},
5111 <bch|pmn> \textless = {100, }, \textgreater = { ,100},
5112 <cmr|pad|ppl|ptm> \textless = {300,100}, \textgreater = {200,100}
5113 <pmn> \textvisiblespace = {100,100}
5114 }
5115
5116 <*m-t|ptm>
5117 \SetProtrusion
5118 <m-t> [ name = QX-it-default,
5119 <ptm> [ name = ptm-it-QX,
5120 <m-t> load = OT1-it ]
5121 <ptm> load = ptm-it ]
5122 { encoding = {QX},
5123 <ptm> family = {ptm,ptmx,ptmj},
5124 shape = {it,sl} }
5125 {
5126 <ptm> 009 = { , 50}, % fk
5127 {=} = {100,100},
5128 <m-t> \textunderscore = {100,100},
5129 <ptm> \textunderscore = {100,150},
5130 \textbackslash = {100,200},
5131 \quotedblbase = {300,400},
5132 <m-t> \guillemotleft = {300,300}, \guillemotright = {300,300},
5133 <ptm> \guillemotleft = {200,400}, \guillemotright = {200,400},
5134 \textexclamdown = {200, }, \textquestiondown = {200, },
5135 \textbraceleft = {200,100}, \textbraceright = {200,200},
5136 \textless = {100,100}, \textgreater = {100,100},
5137 \textminus = {200,200}, \textdegree = {300,150},
5138 <m-t> \copyright = {100,100}, \textregistered = {100,100}
5139 <ptm> \textregistered = {100,150}, \copyright = {100,150},
5140 <ptm> \textDelta = { 70, }, \textdelta = { , 50},
5141 <ptm> \textpi = { 50, 80}, \textmu = { , 80},
5142 <ptm> \texteuro = {200, }, \textellipsis = {100,200},
5143 <ptm> \textquoteleft = {500,400}, \textquoteright = {500,400},
5144 <ptm> \textquotedblleft = {500,300}, \textquotedblright = {400,400},
5145 <ptm> \textapprox = { 50, 50}, \textinfty = {100,100},
5146 <ptm> \textdagger = {150,150}, \textdaggerdbl = {100,100},
5147 <ptm> \textdiv = {150,150}, \textasciitilde = { 80, 80},
5148 <ptm> \texttimes = {100,150}, \textpm = { 50, 80},
5149 <ptm> \textbullet = {300,100}, \textperiodcentered = {300,300},
5150 <ptm> \textquotesingle = {500,500}, \textquotedbl = {300,300},
5151 <ptm> \textperthousand = { ,50}
5152 }
5153
5154 </m-t|ptm>
5155 <*cmr|bch>
5156 \SetProtrusion
5157 <cmr> [ name = cmr-it-T5,
5158 <cmr> load = cmr-it ]
5159 <bch> [ name = bch-it-T5,
5160 <bch> load = bch-it ]
5161 { encoding = T5,
5162 <bch> family = bch,
5163 <cmr> family = cmr,
5164 shape = it }
5165 {
5166 <bch> _ = { ,100},
5167 <cmr> _ = {100,200},
5168 <bch> \textbackslash = {150,150},
5169 <cmr> \textbackslash = {300,300},
5170 <bch> \quotesinglbase = {200,500}, \quotedblbase = {150,500},

```

```

5171 <cmr> \quotesinglbase = {300,700}, \quotedblbase = {200,600},
5172 <bch> \guilsinglleft = {300,400}, \guilsinglright = {200,500},
5173 <cmr> \guilsinglleft = {500,300}, \guilsinglright = {400,400},
5174 <bch> \guillemotleft = {200,300}, \guillemotright = {150,400},
5175 <cmr> \guillemotleft = {400,100}, \guillemotright = {200,300},
5176 <bch> \textbraceleft = {200, }, \textbraceright = { ,200},
5177 <cmr> \textbraceleft = {400,100}, \textbraceright = {200,200},
5178 <bch> \textless = {100, }, \textgreater = { ,100}
5179 <cmr> \textless = {300,100}, \textgreater = {200,100}
5180 }
5181
5182 </cmr|bch>

```

Slanted is very similar to italic.

```

5183 <*cmr>
5184 \SetProtrusion
5185 [ name = cmr-sl,
5186   load = cmr-it-OT1 ]
5187 { encoding = {OT1,OT4},
5188   family = cmr,
5189   shape = sl }
5190 {
5191   L = { ,50},
5192   f = { ,-50},
5193   - = {300, },
5194   \textendash = {400, }, \textemdash = {300, }
5195 }
5196
5197 \SetProtrusion
5198 [ name = cmr-sl-T1,
5199   load = cmr-it-T1 ]
5200 { encoding = {T1,LY1},
5201   family = cmr,
5202   shape = sl }
5203 {
5204   L = { ,50},
5205   f = { ,-50},
5206   - = {300, },
5207   \textendash = {400, }, \textemdash = {300, }
5208 }
5209
5210 \SetProtrusion
5211 [ name = cmr-sl-T5,
5212   load = cmr-it-T5 ]
5213 { encoding = T5,
5214   family = cmr,
5215   shape = sl }
5216 {
5217   L = { ,50},
5218   f = { ,-50},
5219   - = {300, },
5220   \textendash = {400, }, \textemdash = {300, }
5221 }
5222
5223 \SetProtrusion
5224 [ name = lmr-it-T1,
5225   load = cmr-it-T1 ]
5226 { encoding = {T1,LY1},
5227   family = lmr,
5228   shape = {it,sl} }
5229 {
5230   \textquotedblleft = { ,200}, \textquotedblright = { ,200},
5231   \quotesinglbase = { ,400}, \quotedblbase = { ,500}

```

5232 }
5233

Oldstyle numerals are slightly different.

```

5234 \SetProtrusion
5235 [ name = cmr(oldstyle)-it,
5236   load = cmr-it-T1 ]
5237 { encoding = T1,
5238   family   = {hfor,cmor},
5239   shape    = {it,sl} }
5240 {
5241   1 = {250, 50},
5242   2 = {150,-100},
5243   3 = {100,-50},
5244   4 = {150,150},
5245   6 = {200,   },
5246   7 = {200, 50},
5247   8 = {150,-50},
5248   9 = {100, 50}
5249 }
5250
5251 </cmr>
5252 < *pmn>
5253 \SetProtrusion
5254 [ name   = pmnx-it,
5255   load   = pmnj-it ]
5256 { encoding = OT1,
5257   family   = pmnx,
5258   shape    = {it,sl} }
5259 {
5260   1 = {100,150}
5261 }
5262
5263 \SetProtrusion
5264 [ name   = pmnx-it-T1,
5265   load   = pmnj-it-T1 ]
5266 { encoding = {T1,LY1},
5267   family   = pmnx,
5268   shape    = {it,sl} }
5269 {
5270   1 = {100,150}
5271 }
5272
5273 </pmn>
5274 < *ptm>
5275 \SetProtrusion
5276 [ name   = ptm-it-LY1,
5277   load   = ptm-it-T1 ]
5278 { encoding = {LY1},
5279   family   = {ptm,ptmx,ptmj},
5280   shape    = {it,sl} }
5281 {
5282   -                               = {100,100},
5283   \texttrademark                  = {100,100},
5284   \textregistered                 = {100,100},
5285   \textcopyright                  = {100,100},
5286   \textdegree                     = {300,100},
5287   \textminus                      = {200,200},
5288   \textellipsis                   = {100,200},
5289   \% \texteuro                    = {   ,   }, % ?
5290   \textcent                       = {100,100},
5291   \textquotesingle                = {500,   },
5292   \textflorin                    = {100, 70},

```



```

5293 \textdagger           = {150,150},
5294 \textdaggerdbl        = {100,100},
5295 \textbullet            = {150,150},
5296 \textonesuperior      = {150,100},
5297 \texttwosuperior      = {150, 50},
5298 \textthreesuperior    = {150, 50},
5299 \textparagraph        = {100,  },
5300 \textperiodcentered    = {500,300},
5301 \textonequarter       = { 50,  },
5302 \textonehalf          = { 50,  },
5303 \textplusminus        = {100,100},
5304 \textmultiply         = {150,150},
5305 \textdivide          = {150,150}
5306 }
5307
5308 </ptm>

```

15.8.3 Small caps

Small caps should inherit the values from their big brothers. Since values are relative to character width, we don't need to adjust them any further (but we have to reset some characters).

```

5309 <*(big|ugm)>
5310 \SetProtrusion
5311 <m-t> [ name      = OT1-sc,
5312 <bch> [ name      = bch-sc,
5313 <cmr> [ name      = cmr-sc-OT1,
5314 <pad> [ name      = pad-sc,
5315 <pmn> [ name      = pmnj-sc,
5316 <ppl> [ name      = ppl-sc,
5317 <ptm> [ name      = ptm-sc,
5318 <m-t>   load      = default ]
5319 <bch>   load      = bch-default ]
5320 <cmr>   load      = cmr-OT1 ]
5321 <pad>   load      = pad-default ]
5322 <pmn>   load      = pmnj-default ]
5323 <ppl>   load      = ppl-default ]
5324 <ptm>   load      = ptm-default ]
5325 <m-t|bch|pad|pmn> { encoding = OT1,
5326 <cmr|ppl|ptm> { encoding = {OT1,OT4},
5327 <bch>   family    = bch,
5328 <cmr>   family    = cmr,
5329 <pad>   family    = {pad,padx,padj},
5330 <pmn>   family    = pmnj,
5331 <ppl>   family    = {ppl,pplx,pplj},
5332 <ptm>   family    = {ptm,ptmx,ptmj},
5333   shape  = sc }
5334 {
5335   a = {50,50},
5336 <cmr|pad|ppl|ptm> \ae = {50,  },
5337 <bch|pmn>         c = {50,  },
5338 <bch|pad|pmn>     d = {  ,50},
5339 <m-t|bch|cmr|pad|pmn|ptm> f = {  ,50},
5340 <bch|pad|pmn>     g = {50,  },
5341 <m-t|cmr|pad|pmn|ppl|ptm> j = {50,  },
5342 <bch>             j = {100,  },
5343 <m-t|bch|cmr|pad|pmn|ppl> l = {  ,50},
5344 <ptm>             l = {  ,80},
5345 <m-t|bch|cmr|pad|pmn|ppl> 013 = {  ,50}, % fl
5346 <ptm>             013 = {  ,80}, % fl
5347 <bch|pad|pmn>     o = {50,50},

```

```

5348 <pad|pmn> \oe = {50, },
5349 <ppl> p = { 0, 0},
5350 <bch|pad|pmn> q = {50,70},
5351 <ppl> q = { 0, },
5352 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5353 t = {50,50},
5354 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5355 <ptm> y = {80,80}
5356 }
5357
5358 \SetProtrusion
5359 <m-t> [ name = Tl-sc,
5360 <bch> [ name = bch-sc-Tl,
5361 <cmr> [ name = cmr-sc-Tl,
5362 <pad> [ name = pad-sc-Tl,
5363 <pmn> [ name = pmnj-sc-Tl,
5364 <ppl> [ name = ppl-sc-Tl,
5365 <ptm> [ name = ptm-sc-Tl,
5366 <m-t> load = Tl-default ]
5367 <bch> load = bch-Tl ]
5368 <cmr> load = cmr-Tl ]
5369 <pad> load = pad-Tl ]
5370 <pmn> load = pmnj-Tl ]
5371 <ppl> load = ppl-Tl ]
5372 <ptm> load = ptm-Tl ]
5373 { encoding = {Tl,Ly1},
5374 <bch> family = bch,
5375 <cmr> family = cmr,
5376 <pad> family = {pad,padx,padj},
5377 <pmn> family = pmnj,
5378 <ppl> family = {ppl,pplx,pplj},
5379 <ptm> family = {ptm,ptmx,ptmj},
5380 shape = sc }
5381 {
5382 a = {50,50},
5383 <cmr|pad|ppl|ptm> \ae = {50, },
5384 <bch|pmn> c = {50, },
5385 <bch|pad|pmn> d = { ,50},
5386 <m-t|bch|cmr|pad|pmn|ptm> f = { ,50},
5387 <bch|pad|pmn> g = {50, },
5388 <m-t|cmr|pad|pmn|ppl|ptm> j = {50, },
5389 <bch> j = {100, },
5390 <m-t|bch|cmr|pad|pmn|ppl> l = { ,50},
5391 <ptm> l = { ,80},
5392 <m-t|bch|cmr|pad|pmn|ppl> 029 = { ,50}, % fl
5393 <ptm> 029 = { ,80}, % fl
5394 <bch|pad|pmn> o = {50,50},
5395 <bch|pad|pmn> \oe = {50, },
5396 <ppl> p = { 0, 0},
5397 <bch|pad|pmn> q = {50,70},
5398 <ppl> q = { 0, },
5399 <m-t|cmr|pad|pmn|ppl|ptm> r = { , 0},
5400 t = {50,50},
5401 <m-t|bch|cmr|pad|pmn|ppl> y = {50,50}
5402 <ptm> y = {80,80}
5403 }
5404
5405 <!!(blg|ugm)>
5406 <*m-t>
5407 \SetProtrusion
5408 [ name = QX-sc,
5409 load = QX-default ]
5410 { encoding = QX,

```

```

5411     shape    = sc }
5412     {
5413         a = {50,50},
5414         f = { ,50},
5415         j = {50, },
5416         l = { ,50},
5417         013 = { ,50}, % fl
5418         r = { , 0},
5419         t = {50,50},
5420         y = {50,50}
5421     }
5422
5423 </m-t>
5424 <*cmr|bch>
5425 \SetProtrusion
5426 <bch> [ name      = bch-sc-T5,
5427 <bch>    load      = bch-T5 ]
5428 <cmr> [ name      = cmr-sc-T5,
5429 <cmr>    load      = cmr-T5 ]
5430 { encoding = T5,
5431 <bch>    family   = bch,
5432 <cmr>    family   = cmr,
5433     shape    = sc }
5434     {
5435         a = {50,50},
5436 <bch>    c = {50, },
5437 <bch>    d = { ,50},
5438         f = { ,50},
5439 <bch>    g = {50, },
5440 <bch>    j = {100, },
5441 <cmr>    j = {50, },
5442         l = { ,50},
5443 <bch>    o = {50,50},
5444 <bch>    q = { 0, },
5445 <cmr>    r = { , 0},
5446         t = {50,50},
5447         y = {50,50}
5448     }
5449
5450 </cmr|bch>
5451 <*pmn>
5452 \SetProtrusion
5453 [ name      = pmnx-sc,
5454   load      = pmnj-sc ]
5455 { encoding = OT1,
5456   family   = pmnx,
5457   shape    = sc }
5458 {
5459     1 = {230,180}
5460 }
5461
5462 \SetProtrusion
5463 [ name      = pmnx-sc-T1,
5464   load      = pmnj-sc-T1 ]
5465 { encoding = {T1,LY1},
5466   family   = pmnx,
5467   shape    = sc }
5468 {
5469     1 = {230,180}
5470 }
5471

```

15.8.4 Italic small caps

Minion provides real small caps in italics. The `slantsc` package calls them `scit`, Philipp Lehman's `fontinstallationguide` suggests `si`.

```

5472 \SetProtrusion
5473 [ name      = pmnj-scit,
5474   load      = pmnj-it   ]
5475 { encoding = OT1,
5476   family   = pmnj,
5477   shape     = {scit,si} }
5478 {
5479   a = {50, },
5480   \ae = { , -50},
5481   b = {20, -50},
5482   c = {50, -50},
5483   d = {20, 0},
5484   e = {20, -50},
5485   f = {10, 0},
5486   012 = {10, -50}, % fi
5487   013 = {10, -50}, % fl
5488   014 = {10, -50}, % ffi
5489   015 = {10, -50}, % ffl
5490   g = {50, -50},
5491   i = {20, -50},
5492   j = {20, 0},
5493   k = {20, },
5494   l = {20, 50},
5495   m = { , -30},
5496   n = { , -30},
5497   o = {50, },
5498   \oe = {50, -50},
5499   p = {20, -50},
5500   q = {50, },
5501   r = {20, 0},
5502   s = {20, -30},
5503   t = {70, },
5504   u = {50, -50},
5505   v = {100, },
5506   w = {100, },
5507   y = {50, },
5508   z = { , -50}
5509 }
5510
5511 \SetProtrusion
5512 [ name      = pmnj-scit-T1,
5513   load      = pmnj-it-T1   ]
5514 { encoding = {T1,LY1},
5515   family   = pmnj,
5516   shape     = {scit,si}    }
5517 {
5518   a = {50, },
5519   \ae = { , -50},
5520   b = {20, -50},
5521   c = {50, -50},
5522   d = {20, 0},
5523   e = {20, -50},
5524   f = {10, 0},
5525   028 = {10, -50}, % fi
5526   029 = {10, -50}, % fl
5527   030 = {10, -50}, % ffi
5528   031 = {10, -50}, % ffl
5529   g = {50, -50},

```

```

5530     i = {20,-50},
5531     188 = {20, 0}, % ij
5532     j = {20, 0},
5533     k = {20, },
5534     l = {20,50},
5535     m = { , -30},
5536     n = { , -30},
5537     o = {50, },
5538     \oe = {50,-50},
5539     p = {20,-50},
5540     q = {50, },
5541     r = {20, 0},
5542     s = {20,-30},
5543     t = {70, },
5544     u = {50,-50},
5545     v = {100, },
5546     w = {100, },
5547     y = {50, },
5548     z = { , -50}
5549 }
5550
5551 \SetProtrusion
5552 [ name      = pmnx-scit,
5553   load      = pmnj-scit ]
5554 { encoding = OT1,
5555   family   = pmnx,
5556   shape     = {scit,si} }
5557 {
5558   1 = {100,150}
5559 }
5560
5561 \SetProtrusion
5562 [ name      = pmnx-scit-T1,
5563   load      = pmnj-scit-T1 ]
5564 { encoding = {T1,LY1},
5565   family   = pmnx,
5566   shape     = {scit,si} }
5567 {
5568   1 = {100,150}
5569 }
5570
5571 </pmn>

```

15.8.5 Text companion

Finally the TS1 encoding. Still quite incomplete for Times and especially Palatino. Anybody?

```

5572 \SetProtrusion
5573 <m-t> [ name      = textcomp ]
5574 <bch> [ name      = bch-textcomp ]
5575 <blg> [ name      = blg-textcomp ]
5576 <cmr> [ name      = cmr-textcomp ]
5577 <pad> [ name      = pad-textcomp ]
5578 <pmn> [ name      = pmn-textcomp ]
5579 <ppl> [ name      = ppl-textcomp ]
5580 <ptm> [ name      = ptm-textcomp ]
5581 <ugm> [ name      = ugm-textcomp ]
5582 <m-t> { encoding = TS1 }
5583 <!m-t> { encoding = TS1,
5584 <bch>   family   = bch }
5585 <blg>   family   = blg }

```

```

5586 <cmr>      family   = cmr }
5587 <pad>      family   = {pad,padx,padj} }
5588 <pmn>      family   = {pmnx,pmnj} }
5589 <ppl>      family   = {ppl,pplx,pplj} }
5590 <ptm>      family   = {ptm,ptmx,ptmj} }
5591 <ugm>      family   = ugm }
5592 {
5593 <blg>      \textquotestraightbase = {400,500},
5594 <cmr>      \textquotestraightbase = {300,300},
5595 <pad|pmn>  \textquotestraightbase = {400,400},
5596 <blg>      \textquotestraightdblbase = {300,400},
5597 <cmr|pmn>  \textquotestraightdblbase = {300,300},
5598 <pad>      \textquotestraightdblbase = {400,400},
5599 <bch|cmr|pad|pmn|ugm> \texttwelvewdash = {200,200},
5600 <bch|cmr|pad|pmn> \textthreequartersemdash = {150,150},
5601 <ugm>      \textthreequartersemdash = {200,200},
5602 <blg>      \textquotesingle = {500,600},
5603 <cmr|pmn>  \textquotesingle = {300,400},
5604 <pad>      \textquotesingle = {400,500},
5605 <ptm>      \textquotesingle = {500,500},
5606 <ugm>      \textquotesingle = {300,500},
5607 <bch|cmr|pmn> \textasteriskcentered = {200,300},
5608 <blg>      \textasteriskcentered = {150,200},
5609 <pad>      \textasteriskcentered = {300,300},
5610 <ugm>      \textasteriskcentered = {100,200},
5611 <pmn>      \textfractionsolidus = {-200,-200},
5612 <cmr>      \textoneoldstyle = {100,100},
5613 <pmn>      \textoneoldstyle = { , 50},
5614 <cmr>      \textthreeoldstyle = { , 50},
5615 <pad|pmn>  \textthreeoldstyle = { 50, },
5616 <cmr>      \textfouroldstyle = { 50, 50},
5617 <pad|pmn>  \textfouroldstyle = { 50, },
5618 <cmr|pad|pmn> \textsevenoldstyle = { 50, 80},
5619 <cmr>      \textlangle = {400, },
5620 <cmr>      \textrangle = { , 400},
5621 <m-t|bch|pmn|ptm> \textminus = {200,200},
5622 <cmr|pad|ppl> \textminus = {300,300},
5623 <blg|ugm> \textminus = {250,300},
5624 <bch|pad|pmn> \textlbrackdbl = {100, },
5625 <blg>      \textlbrackdbl = {200, },
5626 <bch|pad|pmn> \textrbrackdbl = { , 100},
5627 <blg>      \textrbrackdbl = { , 200},
5628 <pmn>      \textasciigrave = {200,500},
5629 <bch|blg|cmr|pad|pmn> \texttildelow = {200,250},
5630 <pmn>      \textasciibreve = {300,400},
5631 <pmn>      \textasciicaron = {300,400},
5632 <pmn>      \textacutedbl = {200,300},
5633 <pmn>      \textgravedbl = {150,300},
5634 <bch|pmn|ugm> \textdagger = { 80, 80},
5635 <blg>      \textdagger = {200,200},
5636 <cmr|pad> \textdagger = {100,100},
5637 <ptm>      \textdagger = {150,150},
5638 <blg>      \textdaggerdbl = {150,150},
5639 <cmr|pad|pmn> \textdaggerdbl = { 80, 80},
5640 <ptm>      \textdaggerdbl = {100,100},
5641 <bch>      \textbardbl = {100,100},
5642 <blg|ugm> \textbardbl = {150,150},
5643 <bch>      \textbullet = {200,200},
5644 <blg>      \textbullet = {400,500},
5645 <cmr|pad|pmn> \textbullet = { , 100},
5646 <ptm>      \textbullet = {150,150},
5647 <ugm>      \textbullet = { 50, 100},
5648 <bch|cmr|pmn> \textcelsius = { 50, },

```

```

5649 <pad> \textcelsius = { 80, },
5650 <bch> \textflorin = { 50, 50},
5651 <blg> \textflorin = {100,100},
5652 <pad|ugm> \textflorin = { ,100},
5653 <pmn> \textflorin = { 50,100},
5654 <ptm> \textflorin = { 50, 70},
5655 <cmr> \textcolonmonetary = { , 50},
5656 <pad|pmn> \textcolonmonetary = { 50, },
5657 <pmn> \textinterrobang = { ,100},
5658 <pmn> \textinterrobangdown = {100, },
5659 <m-t|pad|ptm> \texttrademark = {100,100},
5660 <bch> \texttrademark = {150,150},
5661 <blg|cmr|ppl> \texttrademark = {200,200},
5662 <pmn> \texttrademark = { 50, 50},
5663 <ugm> \texttrademark = {100,150},
5664 <bch|ugm> \textcent = { 50, },
5665 <ptm> \textcent = {100,100},
5666 <bch> \textsterling = { 50, },
5667 <ugm> \textsterling = { , 50},
5668 <bch> \textbrokenbar = {200,200},
5669 <blg> \textbrokenbar = {250,250},
5670 <ugm> \textbrokenbar = {200,300},
5671 <pmn> \textasciidieresis = {300,400},
5672 <m-t|bch|cmr|pad|ptm|ugm> \textcopyright = {100,100},
5673 <pmn> \textcopyright = {100,150},
5674 <ppl> \textcopyright = {200,200},
5675 <bch|cmr|ugm> \textordfeminine = {100,200},
5676 <pad|pmn> \textordfeminine = {200,200},
5677 <bch|cmr|pad|pmn|ugm> \textlnot = {200, },
5678 <blg> \textlnot = {200,100},
5679 <m-t|bch|cmr|pad|ptm|ugm> \textregistered = {100,100},
5680 <pmn> \textregistered = { 50,150},
5681 <ppl> \textregistered = {200,200},
5682 <pmn> \textasciimacron = {150,200},
5683 <m-t|ppl|ptm> \textdegree = {300,300},
5684 <bch> \textdegree = {150,200},
5685 <blg|ugm> \textdegree = {200,200},
5686 <cmr|pad> \textdegree = {400,400},
5687 <pmn> \textdegree = {150,400},
5688 <bch|cmr|pad|pmn|ugm> \textpm = {150,200},
5689 <blg> \textpm = {100,100},
5690 <ptm> \textpm = { 50, 80},
5691 <bch|blg|ugm> \texttwosuperior = {100,200},
5692 <cmr> \texttwosuperior = { 50,100},
5693 <pad|pmn> \texttwosuperior = {200,200},
5694 <ptm> \texttwosuperior = { 50, 50},
5695 <bch|blg|ugm> \textthreesuperior = {100,200},
5696 <cmr> \textthreesuperior = { 50,100},
5697 <pad|pmn> \textthreesuperior = {200,200},
5698 <ptm> \textthreesuperior = { 50, 50},
5699 <pmn> \textasciicute = {300,400},
5700 <bch|ugm> \textmu = { ,100},
5701 <bch|pad|pmn> \textparagraph = { ,100},
5702 <bch|cmr|pad|pmn> \textperiodcentered = {300,400},
5703 <blg> \textperiodcentered = {400,500},
5704 <ptm> \textperiodcentered = {300,300},
5705 <ugm> \textperiodcentered = {200,500},
5706 <bch|blg|ugm> \textonesuperior = {200,300},
5707 <cmr|pad|pmn> \textonesuperior = {200,200},
5708 <ptm> \textonesuperior = {100,100},
5709 <bch|pad|pmn|ugm> \textordmasculine = {200,200},
5710 <blg|cmr> \textordmasculine = {100,200},
5711 <bch|cmr|pmn> \texteuro = {100, },

```

```

5712 <pad>      \texteuro           = { 50,100},
5713 <bch>      \texttimes           = {200,200},
5714 <blg|ptm>  \texttimes           = {100,100},
5715 <cmr>      \texttimes           = {150,250},
5716 <pad>      \texttimes           = {100,150},
5717 <pmn>      \texttimes           = { 70,100},
5718 <ugm>      \texttimes           = {200,300},
5719 <bch|pad|pmn> \textdiv          = {150,200}
5720 <blg>      \textdiv            = {100,100}
5721 <cmr>      \textdiv            = {150,250}
5722 <ptm>      \textdiv            = { 50,100},
5723 <ugm>      \textdiv            = {200,300},
5724 <ptm>      \textperthousand    = {   ,50}
5725 <ugm>      \textsection         = {   ,100},
5726 <ugm>      \textonehalf        = { 50,100},
5727 <ugm>      \textonequarter     = { 50,100},
5728 <ugm>      \textthreequarters  = { 50,100},
5729 <ugm>      \textsurd           = {   ,100}

```

Remaining slots in the source file.

```

5730 }
5731
5732 <*cmr|pad|pmn|ugm>
5733 \SetProtrusion
5734 <cmr> [ name = cmr-textcomp-it ]
5735 <pad> [ name = pad-textcomp-it ]
5736 <pmn> [ name = pmn-textcomp-it ]
5737 <ugm> [ name = ugm-textcomp-it ]
5738 { encoding = TS1,
5739 <cmr> family = cmr,
5740 <pad> family = {pad,padx,padj},
5741 <pmn> family = {pmnx,pmnj},
5742 <ugm> family = ugm,
5743 <!ugm> shape = {it,sl} }
5744 <ugm> shape = it }
5745 {
5746 <cmr> \textquotestraightbase = {300,600},
5747 <pad|pmn> \textquotestraightbase = {400,400},
5748 <cmr> \textquotestraightdblbase = {300,600},
5749 <pad> \textquotestraightdblbase = {300,400},
5750 <pmn> \textquotestraightdblbase = {300,300},
5751 \texttwelveudash = {200,200},
5752 <cmr|pad|pmn> \textthreequartersemdash = {150,150},
5753 <ugm> \textthreequartersemdash = {200,200},
5754 <cmr> \textquotesingle = {600,300},
5755 <pad> \textquotesingle = {800,100},
5756 <pmn> \textquotesingle = {300,200},
5757 <ugm> \textquotesingle = {500,500},
5758 <cmr> \textasteriskcentered = {300,200},
5759 <pad> \textasteriskcentered = {500,100},
5760 <pmn> \textasteriskcentered = {200,300},
5761 <ugm> \textasteriskcentered = {300,150},
5762 <pmn> \textfractionsolidus = {-200,-200},
5763 <cmr> \textoneoldstyle = {100, 50},
5764 <pad> \textoneoldstyle = {100, },
5765 <pmn> \textoneoldstyle = { 50, },
5766 <pad> \texttwooldstyle = { 50, },
5767 <pmn> \texttwooldstyle = {-50, },
5768 <cmr> \textthreeoldstyle = {100, 50},
5769 <pmn> \textthreeoldstyle = {-100, },
5770 <cmr> \textfouroldstyle = { 50, 50},
5771 <pad> \textfouroldstyle = { 50,100},
5772 <cmr> \textsevenoldstyle = { 50, 80},

```


5773	<i><pad></i>	<code>\textsevenoldstyle</code>	= { 50, },
5774	<i><pmn></i>	<code>\textsevenoldstyle</code>	= { 20, },
5775	<i><cmr></i>	<code>\textlangle</code>	= {400, },
5776	<i><cmr></i>	<code>\textrangle</code>	= { ,400},
5777	<i><cmr pad></i>	<code>\textminus</code>	= {300,300},
5778	<i><pmn></i>	<code>\textminus</code>	= {200,200},
5779	<i><ugm></i>	<code>\textminus</code>	= {250,300},
5780	<i><pad pmn></i>	<code>\textlbrackdbl</code>	= {100, },
5781	<i><pad pmn></i>	<code>\textrbrackdbl</code>	= { ,100},
5782	<i><pmn></i>	<code>\textasciigrave</code>	= {300,300},
5783	<i><cmr pad pmn></i>	<code>\texttildelow</code>	= {200,250},
5784	<i><pmn></i>	<code>\textasciibreve</code>	= {300,300},
5785	<i><pmn></i>	<code>\textasciicaron</code>	= {300,300},
5786	<i><pmn></i>	<code>\textacutedbl</code>	= {200,300},
5787	<i><pmn></i>	<code>\textgravedbl</code>	= {150,300},
5788	<i><cmr></i>	<code>\textdagger</code>	= {100,100},
5789	<i><pad></i>	<code>\textdagger</code>	= {200,100},
5790	<i><pmn></i>	<code>\textdagger</code>	= { 80, 50},
5791	<i><ugm></i>	<code>\textdagger</code>	= { 80, 80},
5792	<i><cmr pad></i>	<code>\textdaggerdbl</code>	= { 80, 80},
5793	<i><pmn></i>	<code>\textdaggerdbl</code>	= { 80, 50},
5794	<i><ugm></i>	<code>\textbardbl</code>	= {150,150},
5795	<i><cmr></i>	<code>\textbullet</code>	= {200,100},
5796	<i><pad></i>	<code>\textbullet</code>	= {300, },
5797	<i><pmn></i>	<code>\textbullet</code>	= { 30, 70},
5798	<i><ugm></i>	<code>\textbullet</code>	= { 50,100},
5799	<i><cmr></i>	<code>\textcelsius</code>	= {100, },
5800	<i><pad></i>	<code>\textcelsius</code>	= {200, },
5801	<i><pmn></i>	<code>\textcelsius</code>	= { 50,-50},
5802	<i><pad></i>	<code>\textflorin</code>	= {100, },
5803	<i><pmn></i>	<code>\textflorin</code>	= { 50,100},
5804	<i><ugm></i>	<code>\textflorin</code>	= { ,100},
5805	<i><cmr></i>	<code>\textcolonmonetary</code>	= {150, },
5806	<i><pad></i>	<code>\textcolonmonetary</code>	= {100, },
5807	<i><pmn></i>	<code>\textcolonmonetary</code>	= { 50,-50},
5808	<i><cmr pad></i>	<code>\texttrademark</code>	= {200, },
5809	<i><pmn></i>	<code>\texttrademark</code>	= { 50,100},
5810	<i><ugm></i>	<code>\texttrademark</code>	= {150, 50},
5811	<i><ugm></i>	<code>\textcent</code>	= { 50, },
5812	<i><ugm></i>	<code>\textsterling</code>	= { , 50},
5813	<i><ugm></i>	<code>\textbrokenbar</code>	= {200,300},
5814	<i><pmn></i>	<code>\textasciidieresis</code>	= {300,200},
5815	<i><cmr></i>	<code>\textcopyright</code>	= {100, },
5816	<i><pad></i>	<code>\textcopyright</code>	= {200,100},
5817	<i><pmn></i>	<code>\textcopyright</code>	= {100,150},
5818	<i><ugm></i>	<code>\textcopyright</code>	= {300, },
5819	<i><cmr></i>	<code>\textordfeminine</code>	= {100,100},
5820	<i><pmn></i>	<code>\textordfeminine</code>	= {200,200},
5821	<i><ugm></i>	<code>\textordfeminine</code>	= {100,200},
5822	<i><cmr pad></i>	<code>\textlnot</code>	= {300, },
5823	<i><pmn ugm></i>	<code>\textlnot</code>	= {200, },
5824	<i><cmr></i>	<code>\textregistered</code>	= {100, },
5825	<i><pad></i>	<code>\textregistered</code>	= {200,100},
5826	<i><pmn></i>	<code>\textregistered</code>	= { 50,150},
5827	<i><ugm></i>	<code>\textregistered</code>	= {300, },
5828	<i><pmn></i>	<code>\textasciimacron</code>	= {150,200},
5829	<i><cmr pad></i>	<code>\textdegree</code>	= {500,100},
5830	<i><pmn></i>	<code>\textdegree</code>	= {150,150},
5831	<i><ugm></i>	<code>\textdegree</code>	= {300,200},
5832	<i><cmr></i>	<code>\textpm</code>	= {150,100},
5833	<i><pad></i>	<code>\textpm</code>	= {200,150},
5834	<i><pmn ugm></i>	<code>\textpm</code>	= {150,200},
5835	<i><cmr></i>	<code>\textonesuperior</code>	= {400, },

```

5836 <pad> \textonesuperior = {300,100},
5837 <pmn> \textonesuperior = {200,100},
5838 <ugm> \textonesuperior = {300,300},
5839 <cmr> \texttwosuperior = {400, },
5840 <pad> \texttwosuperior = {300, },
5841 <pmn> \texttwosuperior = {200,100},
5842 <ugm> \texttwosuperior = {300,200},
5843 <cmr> \textthreesuperior = {400, },
5844 <pad> \textthreesuperior = {300, },
5845 <pmn> \textthreesuperior = {200,100},
5846 <ugm> \textthreesuperior = {300,200},
5847 <ugm> \textmu = { ,100},
5848 <pmn> \textasciicute = {300,200},
5849 <cmr> \textparagraph = {200, },
5850 <pmn> \textparagraph = { ,100},
5851 <cmr> \textperiodcentered = {500,500},
5852 <pad|pmn|ugm> \textperiodcentered = {300,400},
5853 <cmr> \textordmasculine = {100,100},
5854 <pmn> \textordmasculine = {200,200},
5855 <ugm> \textordmasculine = {300,200},
5856 <cmr> \texteuro = {200, },
5857 <pad> \texteuro = {100, },
5858 <pmn> \texteuro = {100,-50},
5859 <cmr> \texttimes = {200,200},
5860 <pad> \texttimes = {200,100},
5861 <pmn> \texttimes = { 70,100},
5862 <ugm> \texttimes = {200,300},
5863 <cmr|pad> \textdiv = {200,200},
5864 <pmn> \textdiv = {150,200},
5865 <ugm> \textdiv = {200,300},
5866 <ugm> \textsection = { ,200},
5867 <ugm> \textonehalf = { 50,100},
5868 <ugm> \textonequarter = { 50,100},
5869 <ugm> \textthreequarters = { 50,100},
5870 <ugm> \textsurd = { ,100}
5871 }
5872
5873 </cmr|pad|pmn|ugm>

```

15.8.6 Computer Modern math

Now to the math symbols for Computer Modern Roman. Definitions have been extracted from `fontmath.ltx`. I did not spend too much time fiddling with these settings, so they can surely be improved.

The math font ‘operators’ (also used for the `\mathrm` and `\mathbf` alphabets) is OT1/cmr, which we’ve already set up above. It’s declared as:

```

\DeclareSymbolFont{operators} {OT1}{cmr} {m}{n}
\SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}

```

`\mathit` (OT1/cmr/m/it) is also already set up.

There are (for the moment) no settings for `\mathsf` and `\mathtt`.

Math font ‘letters’ (also used as `\mathnormal`) is declared as:

```

\DeclareSymbolFont{letters} {OML}{cmm} {m}{it}
\SetSymbolFont{letters} {bold}{OML}{cmm} {b}{it}

```

```

5874 <(*cmr)
5875 \SetProtrusion
5876 [ name = cmr-math-letters ]

```

```

5877 { encoding = OML,
5878     family   = cmm,
5879     series    = {m,b},
5880     shape     = it   }
5881 {
5882     A = {100, 50}, % \mathnormal
5883     B = { 50,   },
5884     C = { 50,   },
5885     D = { 50, 50},
5886     E = { 50,   },
5887     F = {100, 50},
5888     G = { 50, 50},
5889     H = { 50, 50},
5890     I = { 50, 50},
5891     J = {150, 50},
5892     K = { 50,100},
5893     L = { 50, 50},
5894     M = { 50,   },
5895     N = { 50,   },
5896     O = { 50,   },
5897     P = { 50,   },
5898     Q = { 50, 50},
5899     R = { 50,   },
5900     S = { 50,   },
5901     T = { 50,100},
5902     U = { 50, 50},
5903     V = {100,100},
5904     W = { 50,100},
5905     X = { 50,100},
5906     Y = {100,100},
5907     f = {100,100},
5908     h = {   ,100},
5909     i = {   , 50},
5910     j = {   , 50},
5911     k = {   , 50},
5912     r = {   , 50},
5913     v = {   , 50},
5914     w = {   , 50},
5915     x = {   , 50},
5916     "0B = { 50,100}, % \alpha
5917     "0C = { 50, 50}, % \beta
5918     "0D = {200,150}, % \gamma
5919     "0E = { 50, 50}, % \delta
5920     "0F = { 50, 50}, % \epsilon
5921     "10 = { 50,150}, % \zeta
5922     "12 = { 50,   }, % \theta
5923     "13 = {   ,100}, % \iota
5924     "14 = {   ,100}, % \kappa
5925     "15 = {100, 50}, % \lambda
5926     "16 = {   , 50}, % \mu
5927     "17 = {   , 50}, % \nu
5928     "18 = {   , 50}, % \xi
5929     "19 = { 50,100}, % \pi
5930     "1A = { 50, 50}, % \rho
5931     "1B = {   ,150}, % \sigma
5932     "1C = { 50,150}, % \tau
5933     "1D = { 50, 50}, % \upsilon
5934     "1F = { 50,100}, % \chi
5935     "20 = { 50, 50}, % \psi
5936     "21 = {   , 50}, % \omega
5937     "22 = {   , 50}, % \varepsilon
5938     "23 = {   , 50}, % \vartheta
5939     "24 = {   , 50}, % \varpi

```

```

5940 "25 = {100, }, % \varrho
5941 "26 = {100,100}, % \varsigma
5942 "27 = { 50, 50}, % \varphi
5943 "28 = {100,100}, % \leftharpoonup
5944 "29 = {100,100}, % \leftharpoondown
5945 "2A = {100,100}, % \rightharpoonup
5946 "2B = {100,100}, % \rightharpoondown
5947 "2C = {300,200}, % \lhook
5948 "2D = {200,300}, % \rhook
5949 "2E = { ,100}, % \triangleright
5950 "2F = {100, }, % \triangleleft
5951 "3A = { ,500}, % ., \ldotp
5952 "3B = { ,500}, % ,
5953 "3C = {200,100}, % <
5954 "3D = {300,400}, % /
5955 "3E = {100,200}, % >
5956 "3F = {200,200}, % \star
5957 "5B = { ,100}, % \flat
5958 "5E = {200,200}, % \smile
5959 "5F = {200,200}, % \frown
5960 "7C = {100, }, % \jmath
5961 "7D = { ,100} % \wp

```

Remaining slots in the source file.

```

5962 }
5963

```

Math font ‘symbols’ (also used for the `\mathcal` alphabet) is declared as:

```

\DeclareSymbolFont{symbols} {OMS}{cmsy}{m}{n}
\SetSymbolFont{symbols} {bold}{OMS}{cmsy}{b}{n}

```

```

5964 \SetProtrusion
5965 [ name = cmr-math-symbols ]
5966 { encoding = OMS,
5967   family = cmsy,
5968   series = {m,b},
5969   shape = n }
5970 {
5971   A = {150, 50}, % \mathcal
5972   C = { ,100},
5973   D = { , 50},
5974   F = { 50,150},
5975   I = { ,100},
5976   J = {100,150},
5977   K = { ,100},
5978   L = {100, },
5979   M = { 50, 50},
5980   N = { 50,100},
5981   P = { , 50},
5982   Q = { 50, },
5983   R = { , 50},
5984   T = { 50,150},
5985   V = { 50, 50},
5986   W = { , 50},
5987   X = {100,100},
5988   Y = {100, },
5989   Z = {100,150},
5990   "00 = {300,300}, % -
5991   "01 = { ,700}, % \cdotp \cdotp
5992   "02 = {150,250}, % \times
5993   "03 = {150,250}, % *, \ast
5994   "04 = {200,300}, % \div

```

```

5995 "05 = {150,250}, % \diamond
5996 "06 = {200,200}, % \pm
5997 "07 = {200,200}, % \mp
5998 "08 = {100,100}, % \oplus
5999 "09 = {100,100}, % \ominus
6000 "0A = {100,100}, % \otimes
6001 "0B = {100,100}, % \oslash
6002 "0C = {100,100}, % \odot
6003 "0D = {100,100}, % \bigcirc
6004 "0E = {100,100}, % \circ
6005 "0F = {100,100}, % \bullet
6006 "10 = {100,100}, % \asymp
6007 "11 = {100,100}, % \equiv
6008 "12 = {200,100}, % \subseteq
6009 "13 = {100,200}, % \supseteq
6010 "14 = {200,100}, % \leq
6011 "15 = {100,200}, % \geq
6012 "16 = {200,100}, % \preceq
6013 "17 = {100,200}, % \succeq
6014 "18 = {200,200}, % \sim
6015 "19 = {150,150}, % \approx
6016 "1A = {200,100}, % \subset
6017 "1B = {100,200}, % \supset
6018 "1C = {200,100}, % \ll
6019 "1D = {100,200}, % \gg
6020 "1E = {300,100}, % \prec
6021 "1F = {100,300}, % \succ
6022 "20 = {100,200}, % \leftarrow
6023 "21 = {200,100}, % \rightarrow
6024 "22 = {100,100}, % \uparrow
6025 "23 = {100,100}, % \downarrow
6026 "24 = {100,100}, % \leftrightarrow
6027 "25 = {100,100}, % \nearrow
6028 "26 = {100,100}, % \searrow
6029 "27 = {100,100}, % \simeq
6030 "28 = {100,100}, % \Leftarrow
6031 "29 = {100,100}, % \Rightarrow
6032 "2A = {100,100}, % \Uparrow
6033 "2B = {100,100}, % \Downarrow
6034 "2C = {100,100}, % \Leftrightarrow
6035 "2D = {100,100}, % \nrightarrow
6036 "2E = {100,100}, % \swarrow
6037 "2F = { ,100}, % \propto
6038 "30 = { ,400}, % \prime
6039 "31 = {100,100}, % \infty
6040 "32 = {150,100}, % \in
6041 "33 = {100,150}, % \ni
6042 "34 = {100,100}, % \triangle, \bigtriangleup
6043 "35 = {100,100}, % \bigtriangledown
6044 "38 = { ,100}, % \forall
6045 "39 = {100, }, % \exists
6046 "3A = {200, }, % \neg
6047 "3E = {200,200}, % \top
6048 "3F = {200,200}, % \bot, \perp
6049 "5E = {100,200}, % \wedge
6050 "5F = {100,200}, % \vee
6051 "60 = { ,300}, % \vdash
6052 "61 = {300, }, % \dashv
6053 "62 = {100,100}, % \lfloor
6054 "63 = {100,100}, % \rfloor
6055 "64 = {100,100}, % \lceil
6056 "65 = {100,100}, % \rceil
6057 "66 = {150, }, % \lbrace

```

```

6058 "67 = { ,150}, % \rbrace
6059 "68 = {400, }, % \langle
6060 "69 = { ,400}, % \rangle
6061 "6C = {100,100}, % \updownarrow
6062 "6D = {100,100}, % \Updownarrow
6063 "6E = {100,300}, % \, \backslash, \setminus
6064 "72 = {100,100}, % \nabla
6065 "79 = {200,200}, % \dagger
6066 "7A = {100,100}, % \ddagger
6067 "7B = {100, }, % \mathparagraph
6068 "7C = {100,100}, % \clubsuit
6069 "7D = {100,100}, % \diamondsuit
6070 "7E = {100,100}, % \heartsuit
6071 "7F = {100,100} % \spadesuit

```

Remaining slots in the source file.

```

6072 }
6073

```

We don't bother about 'largsymbols', since it will only be used in display math, where protrusion doesn't work anyway. It's declared as:

```
\DeclareSymbolFont{largsymbols}{OMX}{cmex}{m}{n}
```

```

6074 </cmr>
6075 </cfg-t>

```

15.8.7 AMS symbols

Settings for the AMS math fonts (amssymb).

```
6076 <*cfg-u>
```

Symbol font 'a'.

```

6077 <*msa>
6078 \SetProtrusion
6079 [ name = AMS-a ]
6080 { encoding = U,
6081   family = msa }
6082 {
6083   "05 = {150,250}, % \centerdot
6084   "06 = {100,100}, % \lozenge
6085   "07 = { 50, 50}, % \blacklozenge
6086   "08 = { 50, 50}, % \circlearrowright
6087   "09 = { 50, 50}, % \circlearrowleft
6088   "0A = {100,100}, % \rightleftharpoons
6089   "0B = {100,100}, % \leftrightharpoons
6090   "0D = {-50,200}, % \Vdash
6091   "0E = {-50,200}, % \Vvdash
6092   "0F = {-70,150}, % \vdash
6093   "10 = {100,150}, % \twoheadrightarrow
6094   "11 = {100,150}, % \twoheadleftarrow
6095   "12 = { 50,100}, % \leftleftarrows
6096   "13 = { 50, 80}, % \rightrightarrows
6097   "14 = {120,120}, % \upuparrows
6098   "15 = {120,120}, % \downdownarrows
6099   "16 = {200,200}, % \upharpoonright
6100   "17 = {200,200}, % \downharpoonright
6101   "18 = {200,200}, % \upharpoonleft
6102   "19 = {200,200}, % \downharpoonleft
6103   "1A = { 80,100}, % \rightarrowtail
6104   "1B = { 80,100}, % \leftarrowtail

```

```

6105 "1C = { 50, 50}, % \leftrightharrows
6106 "1D = { 50, 50}, % \rightleftarrows
6107 "1E = {250, }, % \Lsh
6108 "1F = { ,250}, % \Rsh
6109 "20 = {100,100}, % \rightsquigarrow
6110 "21 = {100,100}, % \leftrightsquigarrow
6111 "22 = {100, 50}, % \looparrowleft
6112 "23 = { 50,100}, % \looparrowright
6113 "24 = { 50, 80}, % \circeq
6114 "25 = { ,100}, % \succsim
6115 "26 = { ,100}, % \gtrsim
6116 "27 = { ,100}, % \gtrapprox
6117 "28 = {150, 50}, % \multimap
6118 "2B = {100,150}, % \doteqdot
6119 "2C = {100,150}, % \triangleq
6120 "2D = {100, 50}, % \precsim
6121 "2E = {100, 50}, % \lesssim
6122 "2F = { 50, 50}, % \lessapprox
6123 "30 = {100, 50}, % \eqslantless
6124 "31 = { 50, 50}, % \eqslantgtr
6125 "32 = {100, 50}, % \curlyeqprec
6126 "33 = { 50,100}, % \curlyeqsucc
6127 "34 = {100, 50}, % \preccurlyeq
6128 "36 = { 50, }, % \leqslant
6129 "38 = { , 50}, % \backprime
6130 "39 = {250,250}, % \dabar@ : the dash bar in \dash(left,right)arrow
6131 "3C = { 50,100}, % \succcurlyeq
6132 "3E = { , 50}, % \geqslant
6133 "40 = { , 50}, % \sqsubset
6134 "41 = { 50, }, % \sqsupset
6135 "42 = { ,150}, % \vartriangleright, \rhd
6136 "43 = {150, }, % \vartriangleleft, \lhd
6137 "44 = { ,100}, % \trianglerighteq, \unrhd
6138 "45 = {100, }, % \trianglelefteq, \unlhd
6139 "46 = {100,100}, % \bigstar
6140 "48 = { 50, 50}, % \blacktriangledown
6141 "49 = { ,100}, % \blacktriangleright
6142 "4A = {100, }, % \blacktriangleleft
6143 "4B = { ,150}, % \dashrightarrow (the arrow)
6144 "4C = {150, }, % \dashleftarrow
6145 "4D = { 50, 50}, % \vartriangle
6146 "4E = { 50, 50}, % \blacktriangle
6147 "4F = { 50, 50}, % \triangledown
6148 "50 = { 50, 50}, % \eqcirc
6149 "56 = { ,150}, % \Rrightarrow
6150 "57 = {150, }, % \Lleftarrow
6151 "58 = {100,300}, % \checkmark
6152 "5C = { 50, 50}, % \angle
6153 "5D = { 50, 50}, % \measuredangle
6154 "5E = { 50, 50}, % \sphericalangle
6155 "5F = { , 50}, % \varpropto
6156 "60 = {100,100}, % \smallsmile
6157 "61 = {100,100}, % \smallfrown
6158 "62 = { 50, }, % \Subset
6159 "63 = { , 50}, % \Supset
6160 "66 = {150,150}, % \curlywedge
6161 "67 = {150,150}, % \curlyvee
6162 "68 = { 50,150}, % \leftthreetimes
6163 "69 = {100, 50}, % \rightthreetimes
6164 "6C = { 50, 50}, % \bumpeq
6165 "6D = { 50, 50}, % \Bumpeq
6166 "6E = {100, }, % \lll
6167 "6F = { ,100}, % \ggg

```

```

6168 "70 = { 50,100}, % \ulcorner
6169 "71 = {100, 50}, % \urcorner
6170 "75 = {150,200}, % \dotplus
6171 "76 = { 50,100}, % \backsim
6172 "78 = { 50,100}, % \llcorner
6173 "79 = {100, 50}, % \lrcorner
6174 "7C = {100,100}, % \intercal
6175 "7D = { 50, 50}, % \circledcirc
6176 "7E = { 50, 50}, % \circledast
6177 "7F = { 50, 50} % \circledash

```

Remaining slots in the source file.

```

6178 }
6179
6180 </msa>

```

Symbol font ‘b’.

```

6181 <*msb>
6182 \SetProtrusion
6183 [ name = AMS-b ]
6184 { encoding = U,
6185   family = msb }
6186 {
6187   A = { 50, 50}, % \mathbb
6188   C = { 50, 50},
6189   G = { , 50},
6190   L = { , 50},
6191   P = { , 50},
6192   R = { , 50},
6193   T = { , 50},
6194   V = { 50, 50},
6195   X = { 50, 50},
6196   Y = { 50, 50},
6197   "00 = { 50, 50}, % \lvertneqq
6198   "01 = { 50, 50}, % \gvertneqq
6199   "02 = { 50, 50}, % \nleq
6200   "03 = { 50, 50}, % \ngeq
6201   "04 = {100, 50}, % \nless
6202   "05 = { 50,150}, % \ngtr
6203   "06 = {100, 50}, % \nprec
6204   "07 = { 50,150}, % \nsucc
6205   "08 = { 50, 50}, % \lneqq
6206   "09 = { 50, 50}, % \gneqq
6207   "0A = {100,100}, % \nleqslant
6208   "0B = {100,100}, % \ngeqslant
6209   "0C = {100, 50}, % \lneq
6210   "0D = { 50,100}, % \gneq
6211   "0E = {100, 50}, % \npreceq
6212   "0F = { 50,100}, % \nsucceq
6213   "10 = { 50, }, % \precnsim
6214   "11 = { 50, 50}, % \succnsim
6215   "12 = { 50, 50}, % \lnsim
6216   "13 = { 50, 50}, % \gnsim
6217   "14 = { 50, 50}, % \nleqq
6218   "15 = { 50, 50}, % \ngeqq
6219   "16 = { 50, 50}, % \precneqq
6220   "17 = { 50, 50}, % \succneqq
6221   "18 = { 50, 50}, % \precnapprox
6222   "19 = { 50, 50}, % \succnapprox
6223   "1A = { 50, 50}, % \lnapprox
6224   "1B = { 50, 50}, % \gnapprox
6225   "1C = {150,200}, % \nsim
6226   "1D = { 50, 50}, % \ncong

```



```

6227 "1E = {100,150}, % \diagup
6228 "1F = {100,150}, % \diagdown
6229 "20 = {100, 50}, % \varsubsetneq
6230 "21 = { 50,100}, % \varsupsetneq
6231 "22 = {100, 50}, % \subsetneqq
6232 "23 = { 50,100}, % \supsetneqq
6233 "24 = {100, 50}, % \subsetneqq
6234 "25 = { 50,100}, % \supsetneqq
6235 "26 = {100, 50}, % \varsubsetneqq
6236 "27 = { 50,100}, % \varsupsetneqq
6237 "28 = {100, 50}, % \subsetneq
6238 "29 = { 50,100}, % \supsetneq
6239 "2A = {100, 50}, % \subseteq
6240 "2B = { 50,100}, % \supseteq
6241 "2C = { 50,100}, % \parallel
6242 "2D = {100,150}, % \mid
6243 "2E = {150,150}, % \shortmid
6244 "2F = {100,100}, % \shortparallel
6245 "30 = { ,150}, % \vdash
6246 "31 = { ,150}, % \Vdash
6247 "32 = { ,100}, % \nVDash
6248 "33 = { ,100}, % \nVDash
6249 "34 = { ,100}, % \ntrianglerighteq
6250 "35 = {100, }, % \trianglelefteq
6251 "36 = {100, }, % \triangleleft
6252 "37 = { ,100}, % \triangleright
6253 "38 = {100,200}, % \leftarrow
6254 "39 = {100,200}, % \rightarrow
6255 "3A = {100,100}, % \Leftarrow
6256 "3B = { 50,100}, % \Rightarrow
6257 "3C = {100,100}, % \Leftrightarrow
6258 "3D = {100,200}, % \leftrightharrow
6259 "3E = { 50, 50}, % \divideontimes
6260 "3F = { 50, 50}, % \varnothing
6261 "60 = {200, }, % \Finv
6262 "61 = { , 50}, % \Game
6263 "68 = {100,100}, % \eqsim
6264 "69 = { 50, }, % \beth
6265 "6A = { 50, }, % \gimel
6266 "6B = {150, }, % \daleth
6267 "6C = {200, }, % \lessdot
6268 "6D = { ,200}, % \gtrdot
6269 "6E = {100,200}, % \ltimes
6270 "6F = {150,100}, % \rtimes
6271 "70 = { 50,100}, % \shortmid
6272 "71 = { 50, 50}, % \shortparallel
6273 "72 = {200,300}, % \smallsetminus
6274 "73 = {100,200}, % \thicksim
6275 "74 = { 50,100}, % \thickapprox
6276 "75 = { 50, 50}, % \approx
6277 "76 = { 50,100}, % \succapprox
6278 "77 = { 50, 50}, % \precapprox
6279 "78 = {100,100}, % \curvearrowleft
6280 "79 = { 50,150}, % \curvearrowright
6281 "7A = { 50,200}, % \digamma
6282 "7B = {100, 50}, % \varkappa
6283 "7F = {200, } % \backepsilon

```

Remaining slots in the source file.

```

6284 }
6285
6286 (/msb)

```

15.8.8 Euler

Euler Roman font (package `euler`).

```

6287 < *eur>
6288 \SetProtrusion
6289 [ name      = euler ]
6290 { encoding = U,
6291   family   = eur  }
6292 {
6293   "01 = {100,100},
6294   "03 = {100,150},
6295   "06 = {   ,100},
6296   "07 = {100,150},
6297   "08 = {100,100},
6298   "0A = {100,100},
6299   "0B = {   , 50},
6300   "0C = {   ,100},
6301   "0D = {100,100},
6302   "0E = {   ,100},
6303   "0F = {100,100},
6304   "10 = {100,100},
6305   "13 = {   ,100},
6306   "14 = {   ,100},
6307   "15 = {   , 50},
6308   "16 = {   , 50},
6309   "17 = { 50,100},
6310   "18 = { 50,100},
6311   "1A = {   , 50},
6312   "1B = {   , 50},
6313   "1C = { 50,100},
6314   "1D = { 50,100},
6315   "1E = { 50,100},
6316   "1F = { 50,100},
6317   "20 = {   , 50},
6318   "21 = {   , 50},
6319   "22 = { 50,100},
6320   "24 = {   , 50},
6321   "27 = { 50,100},
6322   1   = {100,100},
6323   7   = { 50,100},
6324   "3A = {300,500},
6325   "3B = {200,400},
6326   "3C = {200,100},
6327   "3D = {200,200},
6328   "3E = {100,200},
6329   A   = {   ,100},
6330   D   = {   , 50},
6331   J   = { 50,   },
6332   K   = {   , 50},
6333   L   = {   , 50},
6334   Q   = {   , 50},
6335   T   = { 50,   },
6336   X   = { 50, 50},
6337   Y   = { 50,   },
6338   h   = {   , 50},
6339   k   = {   , 50}
6340 }
6341
```

Extended by the `eulervm` package.

```

6342 \SetProtrusion
6343 [ name      = euler-vm,
```

```

6344     load      = euler ]
6345     { encoding = U,
6346     family     = zeur }
6347     {
6348     "28 = {100,200},
6349     "29 = {100,200},
6350     "2A = {100,150},
6351     "2B = {100,150},
6352     "2C = {200,300},
6353     "2D = {200,300},
6354     "2E = {    ,100},
6355     "2F = {100,   },
6356     "3F = {150,150},
6357     "5B = {    ,100},
6358     "5E = {100,100},
6359     "5F = {100,100},
6360     "80 = {    , 50},
6361     "81 = {200,250},
6362     "82 = {100,200}
6363     }
6364
6365 </eur>

```

Euler Script font (eucal).

```

6366 <*eus>
6367 \SetProtrusion
6368 [ name      = euscript ]
6369 { encoding = U,
6370   family   = eus }
6371 {
6372     A = {100,100},
6373     B = { 50,100},
6374     C = { 50, 50},
6375     D = { 50,100},
6376     E = { 50,100},
6377     F = { 50,   },
6378     G = { 50,   },
6379     H = {    ,100},
6380     K = {    , 50},
6381     L = {    ,150},
6382     M = {    , 50},
6383     N = {    , 50},
6384     O = { 50, 50},
6385     P = { 50, 50},
6386     T = {    ,100},
6387     U = {    , 50},
6388     V = { 50, 50},
6389     W = { 50, 50},
6390     X = { 50, 50},
6391     Y = { 50,   },
6392     Z = { 50,100},
6393     "00 = {250,250},
6394     "18 = {200,200},
6395     "3A = {200,150},
6396     "40 = {    ,100},
6397     "5E = {100,100},
6398     "5F = {100,100},
6399     "66 = { 50,   },
6400     "67 = {    , 50},
6401     "6E = {200,200}
6402 }
6403
6404 \SetProtrusion

```

```
6405 [ name      = euscript-vm,
6406     load      = euscript ]
6407 { encoding = U,
6408   family   = zeus  }
6409 {
6410   "01 = {600,600},
6411   "02 = {200,200},
6412   "03 = {200,200},
6413   "04 = {200,200},
6414   "05 = {150,150},
6415   "06 = {200,200},
6416   "07 = {200,200},
6417   "08 = {100,100},
6418   "09 = {100,100},
6419   "0A = {100,100},
6420   "0B = {100,100},
6421   "0C = {100,100},
6422   "0D = {100,100},
6423   "0E = {150,150},
6424   "0F = {100,100},
6425   "10 = {150,150},
6426   "11 = {100,100},
6427   "12 = {150,100},
6428   "13 = {100,150},
6429   "14 = {150,100},
6430   "15 = {100,150},
6431   "16 = {200,100},
6432   "17 = {100,200},
6433   "19 = {150,150},
6434   "1A = {150,100},
6435   "1B = {100,150},
6436   "1C = {100,100},
6437   "1D = {100,100},
6438   "1E = {250,100},
6439   "1F = {100,250},
6440   "20 = {150,200},
6441   "21 = {150,200},
6442   "22 = {150,150},
6443   "23 = {150,150},
6444   "24 = {100,200},
6445   "25 = {150,150},
6446   "26 = {150,150},
6447   "27 = {100,100},
6448   "28 = {100,100},
6449   "29 = {100,150},
6450   "2A = {100,100},
6451   "2B = {100,100},
6452   "2C = {100,100},
6453   "2D = {150,150},
6454   "2E = {150,150},
6455   "2F = {100,100},
6456   "30 = {100,100},
6457   "31 = {100,100},
6458   "32 = {100,100},
6459   "33 = {100,100},
6460   "34 = {100,100},
6461   "35 = {100,100},
6462   "3E = {150,150},
6463   "3F = {150,150},
6464   "60 = {    ,200},
6465   "61 = {200,   },
6466   "62 = {100,100},
6467   "63 = {100,100},
```

```

6468 "64 = {100,100},
6469 "65 = {100,100},
6470 "68 = {300, },
6471 "69 = { ,300},
6472 "6C = {100,100},
6473 "6D = {100,100},
6474 "6F = {100,100},
6475 "72 = {100,100},
6476 "73 = {200,100},
6477 "76 = { ,100},
6478 "77 = {100, },
6479 "78 = { 50, 50},
6480 "79 = {100,100},
6481 "7A = {100,100},
6482 "7D = {150,150},
6483 "7E = {100,100},
6484 "A8 = {100,100},
6485 "A9 = {100,100},
6486 "AB = {200,200},
6487 "BA = { ,200},
6488 "BB = { ,200},
6489 "BD = {200,200},
6490 "DE = {200,200}
6491 }
6492
6493 </eus>

```

Euler Fraktur font (eufrak).

```

6494 <(*euf)
6495 \SetProtrusion
6496 [ name = mathfrak ]
6497 { encoding = U,
6498   family = euf }
6499 {
6500   A = { , 50},
6501   B = { , 50},
6502   C = { 50, 50},
6503   D = { , 80},
6504   E = { 50, },
6505   G = { , 50},
6506   L = { , 80},
6507   O = { , 50},
6508   T = { , 80},
6509   X = { 80, 50},
6510   Z = { 80, 50},
6511   b = { , 50},
6512   c = { , 50},
6513   k = { , 50},
6514   p = { , 50},
6515   q = { 50, },
6516   v = { , 50},
6517   w = { , 50},
6518   x = { , 50},
6519   1 = {100,100},
6520   2 = { 80, 80},
6521   3 = { 80, 50},
6522   4 = { 80, 50},
6523   7 = { 50, 50},
6524 "12 = {500,500},
6525 "13 = {500,500},
6526   ! = { ,200},
6527   ' = {200,300},
6528   ( = {200, },

```

```

6529      ) = { ,200},
6530      * = {200,200},
6531      + = {200,250},
6532      - = {200,200},
6533      {,} = {300,300},
6534      . = {400,400},
6535      {=} = {200,200},
6536      : = { ,200},
6537      ; = { ,200},
6538      ] = { ,200}
6539  }
6540
6541  </euf>
6542  </cfg-u>

```

15.8.9 Euro symbols

Settings for various Euro symbols (Adobe Euro fonts (packages eurosans, europs), ITC Euro fonts (package euroitc) and marvosym¹⁷).

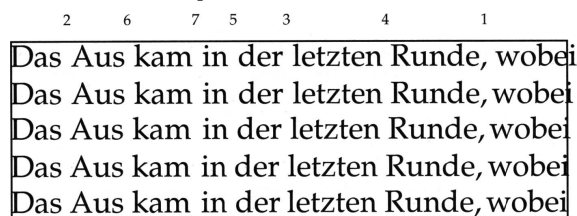
```

6543  <*cfg-e>
6544  \SetProtrusion
6545  <zpeu|euroitc> { encoding = U,
6546  <mvs> { encoding = {OT1,U},
6547  <zpeu> family = zpeu }
6548  <euroitc> family = {euroitc,euroitcs} }
6549  <mvs> family = mvs }
6550  {
6551  <zpeu> E = {50, }
6552  <euroitc> E = {100,50}
6553  <mvs> 164 = {50,50}, % \EUR
6554  <mvs> 068 = {50,-100} % \EURdig
6555  }
6556
6557  <*zpeu|euroitc>
6558  \SetProtrusion
6559  { encoding = U,
6560  <zpeu> family = zpeu,
6561  <euroitc> family = {euroitc,euroitcs},
6562  shape = it* }
6563  {
6564  <zpeu> E = {100,-50}
6565  <euroitc> E = {100,}
6566  }
6567
6568  </zpeu|euroitc>
6569  <*zpeu>
6570  \SetProtrusion
6571  { encoding = U,
6572  family = {zpeus,eurosans} }
6573  {
6574  E = {100,50}
6575  }
6576
6577  \SetProtrusion
6578  { encoding = U,
6579  family = {zpeus,eurosans},
6580  shape = it* }
6581  {
6582  E = {200, }

```

17 Of course, there are many more symbols in this font. Feel free to contribute protrusion settings!

Figure 1: Example for interword spacing (from Siemoneit 1989). The numbers indicate the preference/order when the interword space needs to be shrunk.



```
6583   }
6584
6585 </zpeu>
6586 </cfg-e>
```

15.9 Interword spacing

Default unit is space.

```
6587 <mm-t>
6588 %%% -----
6589 %%% INTERWORD SPACING
6590
6591 \SetExtraSpacing
6592   [ name = default ]
6593   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6594   {
```

These settings are only a first approximation. The following reasoning is from a mail from *Ulrich Dirr*, who also provided the sample in figure 1. I do not claim to have coped with the task.

‘The idea is – analog to the tables for expansion and protrusion – to have tables for optical reduction/expansion of spaces in dependence of the actual character so that the distance between words is optically equal.

When reducing distances the (weighting) order is:

- after commas
- in front of capitals which have optical more room on their left side, e. g., ‘A’, ‘J’, ‘T’, ‘V’, ‘W’, and ‘Y’ [this is not yet possible – RS]
- in front of capitals which have circle/oval shapes on their left side, e. g., ‘C’, ‘G’, ‘O’, and ‘Q’ [ditto – RS]
- after ‘r’ (because of the bigger optical room on the righthand side)
- [before or] after lowercase characters with ascenders

```
6595   { , } = { , -500, 500 },
6596   r = { , -300, 300 },
6597   b = { , -200, 200 },
6598   d = { , -200, 200 },
6599   f = { , -200, 200 },
6600   h = { , -200, 200 },
6601   k = { , -200, 200 },
6602   l = { , -200, 200 },
6603   t = { , -200, 200 },
```

- [before or] after lowercase characters with x-height plus descender with additional optical space, e. g., ‘v’, or ‘w’

```
6604      c = { , -100, 100 },
6605      p = { , -100, 100 },
6606      v = { , -100, 100 },
6607      w = { , -100, 100 },
6608      z = { , -100, 100 },
6609      x = { , -100, 100 },
6610      y = { , -100, 100 },
```

- [before or] after lowercase characters with x-height plus descender without additional optical space

```
6611      i = { , 50, -50 },
6612      m = { , 50, -50 },
6613      n = { , 50, -50 },
6614      u = { , 50, -50 },
```

- after colon and semicolon

```
6615      : = { , 200, -200 },
6616      ; = { , 200, -200 },
```

- after punctuation which ends a sentence, e. g., period, exclamation mark, question mark

```
6617      . = { , 250, -250 },
6618      ! = { , 250, -250 },
6619      ? = { , 250, -250 }
```

The order has to be reversed when enlarging is needed.’

```
6620    }
6621
```

Questions are:

- Is the result really better?
- Is it overdone? (Try with a factor < 1000.)
- Should the first parameter also be used? (Probably.)
- What about quotation marks, parentheses etc.?

Furthermore, there seems to be a pdfTeX bug with spacing in combination with a non-zero \spaceskip (reported by *Axel Berger*):

```
\parfillskip0pt \rightskip0pt plus 1em \spaceskip\fontdimen2\font
test test\par
\pdfadjustinterwordglue2 \stbscode\font`t=-50
test test
\bye
```

15.9.1 Nonfrenchspacing

The following settings simulate \nonfrenchspacing (since space factors will be ignored when spacing adjustment is in effect). They may be used for English contexts.

From the TeXbook:

‘If the space factor f is different from 1000, the interword glue is computed as follows: Take the normal space glue for the current font, and add the extra space if $f \geq 2000$. [...] Then the stretch component is multiplied by $f/1000$, while the shrink component is multiplied by $1000/f$.’

The ‘extra space’ (`\fontdimen7`) for Computer Modern Roman is a third of `\fontdimen2`, i.e., 333.

```
6622 \SetExtraSpacing
6623   [ name      = nonfrench-cmr,
6624     load      = default,
6625     context   = nonfrench ]
6626   { encoding = {OT1,T1,LY1,OT4,QX,T5},
6627     family   = cmr }
6628   {
```

`latex.ltx` has:

```
\def\nonfrenchspacing{
  \sfcode`\.\ 3000
  \sfcode`\? 3000
  \sfcode`\! 3000
```

```
6629   . = {333,2000,-667},
6630   ? = {333,2000,-667},
6631   ! = {333,2000,-667},
```

```
\sfcode`\: 2000
```

```
6632   : = {333,1000,-500},
```

```
\sfcode`\; 1500
```

```
6633   ; = {    , 500,-333},
```

```
\sfcode`\, 1250
```

```
6634   {,}= {    , 250,-200}
```

```
}
```

```
6635   }
6636
```

`fontinst`, however, which is also used to create the PSNFSS font metrics, sets `\fontdimen7` to 240 by default. Therefore, the fallback settings use this value for the first component.

```
6637 \SetExtraSpacing
6638   [ name      = nonfrench-default,
6639     load      = default,
6640     context   = nonfrench ]
6641   { encoding = {OT1,T1,LY1,OT4,QX,T5} }
6642   {
6643     . = {240,2000,-667},
6644     ? = {240,2000,-667},
6645     ! = {240,2000,-667},
6646     : = {240,1000,-500},
6647     ; = {    , 500,-333},
6648     {,}= {    , 250,-200}
6649   }
6650
```

15.10 Additional kerning

Default unit is 1 em.

```
6651 %%% -----
6652 %%% ADDITIONAL KERNING
6653
```

A dummy list to be loaded when no context is active.

```
6654 \SetExtraKerning
6655   [ name = empty ]
6656   { encoding = {OT1,T1,LY1,OT4,QX,T5,TS1} }
6657   { }
6658
```

15.10.1 French

The ratio of `\fontdimen 2` to `\fontdimen 6` varies for different fonts, so that either the kerning of the colon (which should be a space, i. e., `\fontdimen 2`) or that of the other punctuation characters (T_EX's `\thinspace`, i. e., one sixth of `\fontdimen 6`) may be inaccurate, depending on which unit we choose (space or 1em). For Times, for example, a thin space would be 665. I don't know whether French typography really wants a thin space, or rather (as it happens to turn out with CMR) half a space. (Wikipedia¹⁸ claims it should be a quarter of an em, which seems too much to me; then again, it also says that this was a thin space in French typography.)

```
6659 \SetExtraKerning
6660   [ name      = french-default,
6661     context   = french,
6662     unit      = space ]
6663   { encoding = {OT1,T1,LY1} }
6664   {
6665     : = {1000,}, % = \fontdimen2
6666     ; = {500, }, % ~ \thinspace
6667     ! = {500, },
6668     ? = {500, }
6669   }
6670
```

These settings have the disadvantage that a word following a left guillemet will not be hyphenated. This might be fixed in pdfT_EX.

```
6671 \SetExtraKerning
6672   [ name      = french-guillemets,
6673     context   = french-guillemets,
6674     load      = french-default,
6675     unit      = space ]
6676   { encoding = {T1,LY1} }
6677   {
6678     \guillemotleft = { ,800}, % = 0.8\fontdimen2
6679     \guillemotright = {800, }
6680   }
6681
6682 \SetExtraKerning
6683   [ name      = french-guillemets-OT1,
6684     context   = french-guillemets,
6685     load      = french-default,
6686     unit      = space ]
6687   { encoding = OT1 }

```

18 http://fr.wikipedia.org/wiki/Espace_typographique, 5 July 2007.

```
6688 { }
6689
```

15.10.2 Turkish

```
6690 \SetExtraKerning
6691 [ name = turkish,
6692   context = turkish ]
6693 { encoding = {OT1,T1,LY1} }
6694 {
6695   : = {167, }, % = \thinspace
6696   ! = {167, },
6697   {=} = {167, }
6698 }
6699
6700 </m-t>
6701 </config>
```

16 Auxiliary file for micro fine tuning

This file can be used to test protrusion and expansion settings.

```
6702 <*test>
6703 \documentclass{article}
6704
6705 %% Here you can specify the font you want to test, using
6706 %% the commands \fontfamily, \fontseries and \fontshape.
6707 %% Make sure to end all lines with a comment character!
6708 \newcommand*\TestFont{%
6709   \fontfamily{ppl}%
6710   %% \fontseries{b}%
6711   %% \fontshape{it}% sc, sl
6712 }
6713
6714 \usepackage{ifthen}
6715 \usepackage[T1]{fontenc}
6716 \usepackage[latin1]{inputenc}
6717 \usepackage[verbose,expansion=alltext,stretch=50]{microtype}
6718
6719 \pagestyle{empty}
6720 \setlength{\parindent}{0pt}
6721 \newcommand*\crulefill{\cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill}
6722 \newcommand*\testprotrusion[2][ ]{%
6723   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6724   lorem ipsum dolor sit amet,
6725   \ifthenelse{\equal{#1}{r}}{\crulefill}{\leftarrowfill} #2
6726   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6727   you know the rest%
6728   \ifthenelse{\equal{#1}{l}}{\crulefill}{\rightarrowfill}
6729   \linebreak
6730   {\fontencoding{\encodingdefault}%
6731    \fontseries{\seriesdefault}%
6732    \fontshape{\shapedefault}%
6733    \selectfont
6734    Here is the beginning of a line, \dotfill and here is its end}\linebreak
6735 }
6736 \newcommand*\showTestFont{\expandafter\stripprefix\meaning\TestFont}
6737 \def\stripprefix#1>{}
6738 \newcount\charcount
6739 \begin{document}
6740
6741 \microtypesetup{expansion=false}
```

```
6742
6743 {\centering The font in this document is called by:\\
6744 \texttt{\showTestFont}\par}\bigskip
6745
6746 \TestFont\selectfont
6747 This line intentionally left empty\linebreak
6748 %% A -- Z
6749 \charcount=65
6750 \loop
6751 \testprotrusion{\char\charcount}
6752 \advance\charcount 1
6753 \ifnum\charcount < 91 \repeat
6754 %% a -- z
6755 \charcount=97
6756 \loop
6757 \testprotrusion{\char\charcount}
6758 \advance\charcount 1
6759 \ifnum\charcount < 123 \repeat
6760 %% 0 -- 9
6761 \charcount=48
6762 \loop
6763 \testprotrusion{\char\charcount}
6764 \advance\charcount 1
6765 \ifnum\charcount < 58 \repeat
6766 %%
6767 \testprotrusion[r]{,}
6768 \testprotrusion[r]{.}
6769 \testprotrusion[r]{;}
6770 \testprotrusion[r]{:}
6771 \testprotrusion[r]{?}
6772 \testprotrusion[r]{!}
6773 \testprotrusion[l]{\textexclamdown}
6774 \testprotrusion[l]{\textquestiondown}
6775 \testprotrusion[r]{}}
6776 \testprotrusion[l]{({}
6777 \testprotrusion{/}
6778 \testprotrusion{\char`\\}
6779 \testprotrusion{-}
6780 \testprotrusion{\textendash}
6781 \testprotrusion{\textemdash}
6782 \testprotrusion{\textquoteleft}
6783 \testprotrusion{\textquoteright}
6784 \testprotrusion{\textquotedblleft}
6785 \testprotrusion{\textquotedblright}
6786 \testprotrusion{\quotesinglbase}
6787 \testprotrusion{\quotedblbase}
6788 \testprotrusion{\guilsinglleft}
6789 \testprotrusion{\guilsinglright}
6790 \testprotrusion{\guillemotleft}
6791 \testprotrusion{\guillemotright}
6792
6793 \newpage
6794 The following displays the current font stretched by 5%,
6795 normal, and shrunk by 5\%:
6796
6797 \bigskip
6798 \newlength{\MTln}
6799 \newcommand*{\teststring
6800 {ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789}
6801 \settowidth{\MTln}{\teststring}
6802 \microtypesetup{expansion=true}
6803
6804 \parbox{1.05\MTln}{\teststring\linebreak\\
```

```
6805             \teststring}\par\bigskip
6806 \parbox{0.95\MTln}{\teststring}
6807
6808 \end{document}
6809 /test
```

Needless to say that things may always be improved. For suggestions, mail to
w.m.l@gmx.net.

A Change history

Version 1.0 (2004/09/11)

General: Initial version 1

Version 1.1 (2004/09/21)

General: configuration file names in lowercase (suggested by <i>Harald Harders</i>)	79	<code>\MT@get@basefamily</code> : only remove suffix if it is ‘x’ or ‘j’	81
issue an error instead of a warning, when pdfTeX version is too old for autoexpand	123	<code>\MT@get@listname@</code> : don’t check for empty attributes list	81
remove 8-bit characters from the configuration files (suggested by <i>Harald Harders</i>)	131	<code>\MT@ifempty</code> : fix: use category code 12 for the percent character (reported by <i>Tom Kink</i>)	43
Protrusion: add factors for some more characters	136	<code>\MT@is@number</code> : numbers may also be specified in hexadecimal or octal (suggested by <i>Harald Harders</i>)	85
settings for Adobe Minion (contributed by <i>Harald Harders</i>)	137	<code>\MT@pdfTeX@no</code> : fix: version check (reported by <i>Harald Harders</i>)	38
<code>\DeclareCharacterInheritance</code> : new command: possibility to specify character inheritance	107	<code>\MT@permute</code> : don’t use sets for empty encoding ..	109
<code>\MT@declare@sets</code> : remove spaces around set name	94	<code>\MT@split@codes</code> : fix: allow zero and negative values ..	58
<code>\MT@DeclareSet</code> : remove spaces around first argument	94	<code>\MT@use@set</code> : remove spaces around set name	98
<code>\MT@find@file</code> : fix: also check whether the file for the base font family has already been loaded ..	79	<code>\UseMicrotypeSet</code> : remove spaces around first argument	98

Version 1.2 (2004/10/03)

Font sets: declare cmr as an alias of cmr	129	changed	95
new: allmath and basicmath	128	<code>\MT@get@inh@list</code> : fix: set inheritance list \globally to \empty	83
Protrusion: add settings for Computer Modern Roman and Adobe Garamond in TS1 encoding	157	<code>\MT@get@listname@</code> : alternatively check for alias font name	81
add settings for Computer Modern Roman math symbols	162	<code>\MT@get@size</code> : additional magic to catch some errors ..	96
<code>\MT@encoding@check</code> : check whether only one encoding specified	108	hijack \set@fontsize instead of \setfontsize ..	96
<code>\MT@familyalias</code> : define alias font name as an alternative, not as a replacement	55	<code>\MT@loop</code> : fix: new macro, used instead of \loop ..	47
<code>\MT@get@basefamily</code> : also remove ‘w’ (swash capitals)	81	<code>\MT@maybe@do</code> : also check for alias font name	55
<code>\MT@get@highlevel</code> : check whether defaults have		<code>\MT@permute@@@@@</code> : more sanity checks for \SetProtrusion and \SetExpansion	110
		<code>\MT@setupfont</code> : also search for alias font file	53
		fix: call \@@enc@update if necessary	53

Version 1.3 (2004/10/27)

General: fix: specifying load option does no longer require to give a name, too	104	<code>\MT@fix@catcode</code> : check some category codes (compatibility with german)	34
Font sets: declare aer, zer and hfor as aliases of cmr	129	<code>\MT@load@list</code> : check whether list exists	79

Version 1.4 (2004/11/12)

General: check for pdfcprot	50	(OT1, T1, lmr)	142
don’t use scratch registers in global definitions ..	83	<code>\microtypesetup</code> : fix: set the correct levels, and remember them; warning when enabling an option disabled in package options	118
use \pickup@font instead of \define@newfont as the hook for \MT@setupfont	89	<code>\SetExpansion</code> : fix: specifying extra options does no longer require to give a name, too	101
use one instead of five counters	47		
Protrusion: tweak quote characters for cmr variants			

Version 1.4a (2004/11/17)

General: new option: final	115	when reading files (reported by <i>Michael Hoppe</i>)	80
\MT@cfg@catcodes: fix: reset some more catcodes			

Version 1.4b (2004/11/26)

General: fix: set catcodes before reading global configuration file (reported by <i>Christoph Bier</i>)	116	name if encoding failed	56
new message if \pdfoutput is changed	121	\MT@get@basefamily: fix: failed for font names of the form abczz (reported by <i>Georg Verwey</i>)	81
optimisation: use less \expandafters and \csnames	41	\MT@get@slot: don't define \MT@char globally (save stack problem)	83
Protrusion: harmonise dashes in upshape and italic (cmr, pad, ppl)	136	\MT@ifdimen: don't set \MT@count globally (save stack problem)	44
slanted like italics	145	\MT@use@set: don't use undeclared font sets	98
\MT@checklist@family: fix: don't try alias family			

Version 1.5 (2004/12/15)

General: defaults: step: 4 (suggested by <i>Hàn Thế Thành</i>)	115	\MT@cfg@catcodes: reset catcode of '=' (compatibility with Turkish babel)	80
defaults: calculate step as min(stretch,shrink)/5	122	\MT@fix@catcode: reset catcode of '~ (compatibility with chemsym)	34
defaults: turn off expansion for DVI output	121	\MT@get@highlevel: don't test defaults if called after begin document	95
disable automatic expansion for DVI output	122	\MT@scale@factor: warning for factors outside limits	60
new option: selected, by default false (suggested by <i>Hàn Thế Thành</i>)	113	\MT@scale@to@em: don't use \lpcode and \rprcode for the calculation	59
Documentation: add 'Short history'	30	\MT@set@ex@codes: allow non-selected font expansion	65
add note about DVIoutput option	9	\MT@set@pr@codes: adjust protrusion factors before setting the inheriting characters	57
Inheritance: remove \ss from T1 list, add \DJ	131		
Protrusion: settings for Bitstream Charter	137		
\DeclareMicrotypeAlias: remove spaces around arguments	99		

Version 1.6 (2005/01/24)

General: defaults: turn off expansion for old pdfTeX versions	116	improve settings for numbers (pointed out by <i>Peter Muthesius</i>)	139
disable automatic expansion for old pdfTeX versions	123	tune CMR math letters (OML encoding)	162
load a font if none is selected	52	\MT@get@charwd: use e-TeX's \fontcharwd, if available	60
new option: factor, by default 1000	115	\MT@get@inh@list: correct message if selected is false	83
restructure dtx file	128	\MT@set@ex@codes: introduce factor option	65
test whether \pickup@font has changed	91	\MT@set@pr@codes: introduce factor option	57
test whether numeric options receive a number	115	\MT@use@set: retain current set if new set is undeclared	98
use e-TeX's \ifcsname and \ifdefined if defined	42	\MT@vinfo: new macro instead of \ifMT@verbose	35
Protrusion: add italic uppercase Greek letters	145		

Version 1.6a (2005/02/02)

Documentation: add table of fonts with tailored protrusion settings	21	reported by <i>Bernard Gaulle</i>)	83
\MT@get@slot: completely redone, hopefully more robust (compatible with frenchpro; problem		\MT@pdfTeX@no: new macro	37
		\MT@reset@ef@codes: only reset \efcodes for older pdfTeX versions	65

Version 1.7 (2005/03/23)

General: allow specification of size ranges (suggested by <i>Andreas Böhmann</i>)	95	fix: remove space after autoexpand	106
disallow automatic expansion if pdfTeX too old	106	modify \showhyphens	123
		new value for verbose option: errors	115

shorter command names	47	test whether <code>\(encoding)\(...</code> is defined	84
warning when running in draft mode	121	<code>\MT@if@list@exists</code> : don't define <code>\MT@#1@c@name</code>	
Documentation: add hint about compatibility	27	globally, here and elsewhere	82
remove table of match order	12	<code>\MT@if@dimen</code> : comparison with 1 to allow size smaller than 1 (suggested by <i>Andreas Böhmann</i>)	44
Protrusion: fix: remove <code>\</code> from OT1, add <code>\textbackslash</code> to T1 encoding	140	<code>\MT@increment</code> : use e-TeX's <code>\numexpr</code> if available	47
<code>\DeclareMicrotypeAlias</code> : may also be used inside configuration files	99	<code>\MT@is@composite</code> : new macro: construct command for composite character; no uncontrolled expansion	87
<code>\LoadMicrotypeFile</code> : new command (suggested by <i>Andreas Böhmann</i>)	100	<code>\MT@scale</code> : new macro: use e-TeX's <code>\numexpr</code> if available	47
<code>\Microtype@Hook</code> : new command for font package authors	117	<code>\MT@set@ex@codes</code> : two versions of this macro	65
<code>\microtypesetup</code> : fix: warning also when setting to (no)compatibility	118	<code>\MT@split@name</code> : don't define <code>\MT@encoding &c.</code> globally	95
<code>\MT@begin@catcodes</code> : also use inside configuration commands	80	<code>\MT@test@ast</code> : make it simpler	95
<code>\MT@cfg@catcodes</code> : reset catcode of <code>'</code> (compatibility with french* packages)	80	<code>\MT@try@order</code> : always check for size, too (suggested by <i>Andreas Böhmann</i>)	81
<code>\MT@get@listname@</code> : use <code>\etfor</code> (<i>Andreas Böhmann</i> 's idea)	81	fix: also check for <code>//(series)/(shape)//</code> (reported by <i>Andreas Böhmann</i>)	81
<code>\MT@get@slot</code> : remove backslash hack	83	<code>\MT@warn@code@too@large</code> : new macro: type out maximum protrusion factor	61
test for <code>\chardef</code> commands	84	<code>\MT@warn@err</code> : new macro: for verbose=errors	35

Version 1.8 (2005/06/23)

General: <code>\SetProtrusion</code> : new key: unit	105	<code>\MT@find@file</code> : no longer wrap names in commands	79
if font substitution has occurred, set up the substitute font, not the selected one	89	<code>\MT@get@charwd</code> : warning for missing (resp. zero-width) characters	60
new option: config to load a different main configuration file	116	<code>\MT@get@font@dimen@six</code> : new macro: test whether <code>\fontdimen6</code> is defined	58
new option: unit, by default character	116	<code>\MT@get@listname@</code> : made recursive	81
Documentation: add example for factor option	13	<code>\MT@get@slot</code> : fix: expand active characters	84
add example of how to get rid of a widow (suggested by <i>Adam Kucharczyk</i>)	15	test whether <code>\(encoding)\(...</code> is defined made more robust	84
add hint about error messages	27	<code>\MT@get@unit</code> : new macro: get unit for codes	62
Font sets: add U encoding to allmath	128	<code>\MT@in@list</code> : made recursive	46
declare <code>pxr</code> and <code>txr</code> as aliases of <code>ppl</code> resp. <code>ptm</code>	129	<code>\MT@is@active</code> : new macro: translate inputencoding-defined characters	85
Inheritance: remove <code>\DJ</code> from T1 list (it's the same as <code>\DH</code>)	131	<code>\MT@is@letter</code> : warning for non-ASCII characters	85
Protrusion: add LY1 characters for Times	144	<code>\MT@led@kern</code> : character protrusion with <code>ledmac</code>	49
settings for AMS math fonts	166	<code>\MT@map@clist@n</code> : new macro: used instead of <code>\@for</code>	45
verified settings for slanted Computer Modern Roman	151	<code>\MT@map@tlist@n</code> : new macro: used instead of <code>\@tfor</code>	46
<code>\add@accent</code> : fix: disable micro-typographic setup inside <code>\add@accent</code> (reported by <i>Stephan Hennig</i>)	91	<code>\MT@old@cmd</code> : renamed commands from <code>\..MicroType..</code> to <code>\..Microtype..</code>	35
<code>\DeclareMicrotypeAlias</code> : warning when overriding an alias font	99	<code>\MT@pdftex@no</code> : case 5: pdfTeX 1.30	37
<code>\DeclareMicrotypeSetDefault</code> : new command: set default font set	98	<code>\MT@permute@@@@@</code> : add ranges to the beginning of the lists	110
<code>\MT@cfg@catcodes</code> : reset catcodes of the remaining ASCII characters	80	<code>\MT@scale</code> : fix: remove spaces in e-TeX variant (reported by <i>Mark Rossi</i>)	47
<code>\MT@check@rlist</code> : made recursive	111	<code>\MT@setupfont@hook</code> : restore <code>\%</code> and <code>\#</code> when <code>hyperref</code> is loaded	50
<code>\MT@curr@list@name</code> : new macro: current list type and name	87	restore csquotes's active characters	50
<code>\MT@declare@sets</code> : warning when redefining a set	94	restore percent character if Spanish <code>babel</code> is loaded	50
<code>\MT@define@set@key@</code> : use comma lists instead of token lists	94	<code>\MT@split@codes</code> : get character width once only	58
		<code>\MT@use@set</code> : fix: remove braces in first line	98
		<code>\MT@xadd</code> : simplified	45

Version 1.9 (2005/10/28)

General: <code>\DeclareMicrotypeSet</code> : new key: font	97	<code>\SetProtrusion</code> : value 'relative' renamed to	
--	----	---	--

‘character’ for key unit	105	settings for T5 encoded Computer Modern Roman	136
allow context-specific font setup	89	\DisableLigatures: new command: disable ligatures (requires pdf \TeX 1.30)	100
disable expansion if both step and shrink are zero	123	\microtypecontext: new command: change setup context in the document	92
disable microtype setup inside hyperref’s		\MT@checklist@family: fix: add two missing \expandafters	56
\pdfstringdef (reported by <i>Hàn Thế Thành</i>)	51	\MT@detokenize@c: fix the \mathcal{E} - \TeX version	43
fix: use true as the default value	113	\MT@exp@two@n: new macros: less \expandafters	42
option unit: rename value relative to character	116	\MT@get@opt: new key ‘preset’ to set all characters to the specified value before loading the lists	62
warning if user requested zero step	122	\MT@is@active: redone: use \set@display@protect	85
Documentation: add hint about verbatim environment	26	\MT@is@letter: using \catcode should be more efficient than inspecting the \meaning	85
add remark about Type 1 fonts required for automatic font expansion	8	\MT@maybe@do: redone	55
Font sets: add OT4 encoding to text sets	128	\MT@pdf@tex@no: compatibility with \TeX Live hack (reported by <i>Herbert Voß</i>)	38
add T5 encoding to text sets	128	\MT@rem@from@clist: new macro: remove an item from a comma list	46
declare qpl and qtm (qfonts, \TeX Gyre) as aliases of ppl resp. ptm	129	\MT@scale@factor: generalised	60
Inheritance: add list for OT4	132	\MT@toks: use instead of \toks@	39
add list for T5 (requested by <i>Hàn Thế Thành</i>)	134	\SetProtrusion: (et al.) new key: font	101
Protrusion: fix: remove uppercase Greek letters from T1 encoded CMR	141		
settings for OT4 encoding (Computer Modern Roman, Palatino, Times)	136		

Version 1.9a (2005/12/05)

General: ‘(file name)/(line number)’ as default list name	104	diately (requested by <i>Georg Verwey</i>)	94
new option: deferfontsetup, by default true	114	\MT@get@highlevel: no longer check whether defaults have changed	95
remove superfluous test whether \pickup@font has changed	91	\MT@ifdefined@c@T: new macros: true case only	42
Documentation: add explanation for error message in DVI mode	27	\MT@ifint: use \pdfmatch if available	43
add explanation for error message with non-Type 1 fonts	27	\MT@ifstreq: use \pdfstrcmp if available	44
Font sets: declare mdbch (mathdesign) as an alias of Charter	130	\MT@in@clist: fix	46
Protrusion: fix: remove ‘_’ from OT1 encoding	141	\MT@info@missing@char: info instead of warning (after <i>Michael Hoppe</i> reported that the ‘fl’ ligature is missing in Palatino SC)	60
settings for T5 encoded Charter	136	\MT@is@feature: new macro: check for pdf \TeX feature	48
\microtypesetup: inside the preamble, accepts all package options	118	\MT@map@clist@n: following \LaTeX 3	45
\MT@check@font@cx: optimise context-sensitive setup	91	\MT@permute@@@@: don’t define permutations for unused encodings	109
\MT@define@set@key@: don’t expand variables immediately		\MT@rem@from@clist: fix	46
		\MT@setup@: defer setup until the end of the preamble	49

Version 1.9b (2006/01/20)

General: compatibility with listings: sanitise more catcodes (reported by <i>Holger Uhr</i>)	51	add samples of micro-typographic features	4
compatibility with the extendedchar option of the listings package	52	\MT@features: use throughout the package to adjust to beta-ness	48
Documentation: activate expansion in the distributed PDF	1	\MT@ifdimen: use \pdfmatch if available	44
		\MT@warn@code@to@large: fix calculation with present factor	61

Version 1.9c (2006/02/02)

Documentation: add example of how to increase protrusion of footnote markers (suggested by <i>Georg Verwey</i>)	22	\MT@define@code@key@font: fix: context was ignored	103
Protrusion: settings for URW Garamond	137	\MT@define@code@key@size: fix: embrace \MT@tempsize in \csname (bug introduced in v1.9b)	103

Version 1.9d (2006/05/05)

Font sets: md* instead of m series in basic sets	128	\MT@get@font@dimen: warning for zero fontdimen . .	60
add QX encoding to text sets	128	\MT@get@opt: optimise: don't reset when preset op-	
Inheritance: add list for QX encoding (contributed by		tion is set	62
Maciej Eder)	133	set list name before presetting	62
Protrusion: settings for QX encoding (contributed by		\MT@is@active: support for Unicode (inputenc/utf8) .	86
Maciej Eder)	142	\MT@setupfont@hook: restore % and # when tex4ht	
settings for Euro symbols (Adobe, ITC, marvosym)	174	is loaded (reported by <i>Peter Dybala</i>)	50
tweak AMS settings	166	\SetProtrusion: (et al.) optimise: unify keys for	
\DeclareCharacterInheritance: fix: empty context	107	mandatory argument	101
\MT@detokenize@n: new macro: use \detokenize if		(et al.) split keys of optional and mandatory argu-	
available	43	ment	101
\MT@get@ex@opt: fix: evaluate preset	66		

Version 1.9e (2006/07/28)

General: fix: default value for activate: true . . .	113	settings for Euler Roman font	170
Documentation: add hint about unknown encodings	26	\DeclareCharacterInheritance: new key 'inputenc'	
include LPPL	197	to set the input encoding	107
Font sets: declare zeur and zeus (eulervm) as aliases		\MT@rem@from@cllist: model after \@removeelement	46
of eur resp. eus (euler)	129	\MT@setup@: empty \MT@setup@ after use (compatibil-	
Inheritance: adapt to marvosym's changed encoding	135	ity with the combine class)	49
Protrusion: complete settings for Euler Fraktur and		\pickup@font: no tracing with trace package	90
Script fonts	173	\SetExpansion: new key: inputenc	101
fix: forgotten comma in mt-mvs.cfg; adapt to		\SetProtrusion: (et al.) new key: inputenc	101
marvosym's changed encoding	174		

Version 1.9f (2006/09/09)

Protrusion: fix: euler-vm did not load euler settings	170	\MT@reset@context: only reset context if it has actu-	
\MT@curr@list@name: fix: \MessageBreak must not be		ally been changed	92
expanded	87	\MT@set@inh@list: fix: forgotten comma in the fea-	
\MT@gdef@n: new macros: global variants	41	tures list	107
\MT@get@inh@list: fix: input encoding must be set		\MT@set@named@keys: new macro: set name first, sim-	
after the inheritance list has been parsed	83	plify parsing of optional argument	103
\MT@glet: new macro	41	\SetProtrusion: (et al.) set catcodes before parsing	
		optional argument	101

Version 2.0 (2007/01/14)

General: (beta:1) new option: babel, by default false		Documentation: add hint about how to increase	
(language-dependent setup suggested by <i>Ulrich</i>		font_max and font_mem_size	28
<i>Dirr</i>)	113	add hint about warning when tracking and expan-	
(beta:1) new option: letterspace, by default 100	115	sion is applied to a font	28
(beta:3) compatibility with soul: register \textls		add remark about 'draft' option disabling	
and \lsstyle	52	microtype (noted by <i>Michalis Miatidis</i>)	9
(beta:8) option 'babel': fix: switch off French		qualify hint about web documents with regard to	
babel's shorthands properly (reported by <i>Daniel</i>		older pdfTeX versions	25
<i>Flipo</i>)	126	qualify hints about expansion error messages with	
(beta:8) option 'babel': switch off Turkish babel's		regard to older pdfTeX versions	27
shorthands	127	Font sets: add footnotesize and scriptsize sets	128
compatibility with listings: set catcode of back-		add smallcaps set	128
slash to zero (reported by <i>Steven Bath</i>)	51	\DeclareMicrotypeBabelHook: (beta:1) new com-	
maybe disable \MT@no ligatures after the preamble	125	mand: interaction with babel	100
new package letterspace: a stripped-down ver-		\lsstyle: (beta:1) new command: letterspacing . .	72
sion, containing the letterspacing commands		(beta:8) fix: font switches don't pose a problem	
only	1	anymore	72
option 'unit', \SetProtrusion: deprecate value		(beta:8) fix: letterspacing commands may be nes-	
'relative' completely	105	ted	72

totally redone, using the new <code>\letterspacefont</code>	72	<code>\SetExtraKerning</code> : (beta:1) new command: additional kerning	102
<code>\MT@declare@sets</code> : fix: empty size list when redefining set	94	<code>\SetExtraSpacing</code> : (beta:1) new command: adjustment of interword spacing	102
<code>\MT@is@symbol</code> : made even more robust	86	<code>\SetTracking</code> : new command: tracking	102
<code>\MT@load@inputenc</code> : sanitise catcodes before loading input encoding (problem with listings)	63	<code>\textls</code> : (beta:1) new command: letterspacing	77
<code>\MT@pdfTeX@no</code> : (beta:1) case 6: pdfTeX 1.40	37	starred version: remove spaces around text	77
<code>\MT@split@name</code> : adjust to possible letterspacing	55	<code>\tracingmicrotypeinpdf</code> : new debug method: mark all fonts with PDF annotations	36

Version 2.1 (2007/01/21)

General: compatibility with CJK: also check for its definition	89	<code>\lslig</code> : new command: protect ligatures in letterspaced text	73
compatibility with pinyin: disable microtype in <code>\py@macron</code> (reported by <i>Sven Naumann</i>)	52	<code>\MT@get@ls@basefont</code> : redone: use <code>\pdfmatch</code> to make it bullet-proof	73
fix: letterspace package forgot to load <code>keyval</code>	39	<code>\textls</code> : fix: use <code>\hmode@bgroup</code>	77

Version 2.2 (2007/07/14)

General: disable microtype if wordcount is loaded (reported by <i>Ross Hetherington</i>)	48	fix: ϵ -TeX version shouldn't use <code>\x</code> and <code>\y</code> (found by <i>Wiebke Petersen</i>)	44
new option: <code>copyfonts</code>	114	<code>\MT@is@composite</code> : more robust: expand exactly once	87
simplify key declarations	104	<code>\MT@is@symbol</code> : expand once more (for frenchpro)	86
warning if stretch or shrink aren't multiples of step	123	<code>\MT@lsfont</code> : use <code>\font@name</code> , not <code>\MT@font</code>	70
Documentation: add hint about error message with pdfTeX 1.40	27	<code>\MT@maybe@etex</code> : use catcode trickery	41
add hint about extra TOC leader dot (first discovered by <i>Morten Høgholm</i>)	26	<code>\MT@pdfTeX@no</code> : case 7: pdfTeX 1.40.4	38
add overview	6	<code>\MT@preset@aux@space</code> : generalised	63
logo transparency and amusement	1	<code>\MT@requires@luatex</code> : (basic) support for luaTeX	41
Font sets: declare FPL Neu as an alias of Palatino	129	<code>\MT@set@all@pr</code> : (et al.) allow empty values	58
declare <code>chr</code> (<code>chmath</code>) as an alias of Charter (reported by <i>Geoff Vallis</i>)	130	<code>\MT@set@inputenc</code> : only load <code>inputenc</code> files if necessary	63
default set for tracking: <code>smallcaps</code>	129	<code>\MT@set@tr@codes</code> : disable ligatures in letterspaced fonts manually (due to change in pdfTeX 1.40.4)	71
Inheritance: remove <code>'-'</code> \rightarrow <code>'127'</code>	132	possibility to customise interword spacing	70
Protrusion: settings for Bitstream Letter Gothic	137	<code>\MT@setupfont</code> : don't call <code>\@enc@update</code> anymore	53
Spacing: add sample	175	only add features that are available with the respective pdfTeX	53
Tracking: add ligatures that are to be disabled	135	<code>\MT@setupfont@hook</code> : restore percent character if Galician babel is loaded	50
<code>\DeclareMicrotypeVariants</code> : new command	99	<code>\MT@the@pr@code@tr</code> : adjust protrusion of letterspaced fonts	58
<code>\DisableLigatures</code> : new optional argument: disable selected ligatures only	100	<code>\MT@tracking</code> : remember fonts that shouldn't be letterspaced	69
<code>\lslig</code> : always defined	73	<code>\MT@tracking@</code> : fix: tracking couldn't be re-enabled	69
<code>\MT@checklist@font</code> : fix: construct font name from characteristics	57	<code>\MT@warn@tracking@DVI</code> : warning when letterspacing in DVI mode	125
<code>\MT@copy@font</code> : optionally work on copies of fonts	54	<code>\MT@with@babel@and@T</code> : also inspect class options	49
<code>\MT@get@basefamily</code> : redone, working on font names and suffixes of arbitrary length	81	<code>\pickup@font</code> : letterspace: setup inside group	90
<code>\MT@get@charwd</code> : subtract letterspacing amount from width	60	<code>\SetTracking</code> : new key 'no ligatures' to disable ligatures of letterspaced fonts	102
<code>\MT@get@ls@basefont</code> : fix again: remember base font in a macro	73	new keys 'spacing' and 'outer spacing' to adjust interword spacing (suggested by <i>Steven E. Harris</i>)	102
<code>\MT@ifdimen</code> : employ luaTeX features if available	44	third argument may be empty	102
<code>\MT@ifint</code> : employ luaTeX features if available	43	<code>\textmicrotypecontext</code> : new command: wrapper around <code>\microtypecontext</code>	92
<code>\MT@ifstreq</code> : employ luaTeX features if available	44		

Version 2.3 (2007/12/23)

General: disable <code>\microtypecontext</code> in <code>hyperref</code> 's <code>\pdfstringdef</code>	51	<code>\MT@in@clist</code> : fix: don't use <code>\x</code> (reported by <i>Peter Meier</i>)	46
enable protrusion when tracking is enabled ...	124	<code>\MT@is@active</code> : support for extended Unicode (<code>inputenc/utf8x</code> resp. <code>ucs</code>) – experimental ...	86
fix: really switch off Turkish shorthands	127	<code>\MT@noligatures</code> : fix: set evaluation didn't work (bug introduced in 2.2)	78
new value for verbose option: silent (suggested by <i>Karl Berry</i>)	115	<code>\MT@plain</code> : letterspace: support for <code>eplain/miniltx</code>	37
turned some warnings into errors	115	<code>\MT@set@curr@os</code> : adjusting spaces made more reli- able	71
Documentation: add kerning sample	20	<code>\MT@set@tr@codes</code> : also adjust tracking if protrusion is not enabled, and even for letterspace (reported by <i>Stephan Hennig</i>)	71
add letterspacing illustration	17	possibility to customise outer kerning (suggested by <i>Stephan Hennig</i>)	70
<code>\do@subst@correction</code> : remember substitute font for all times (reported by <i>Stephan Hennig</i>)	90	<code>\MT@SetTracking</code> : sanity check for value	102
<code>\lslig</code> : redone: extract outer kerns from current letterspacing amount	73	<code>\MT@tr@outer@l</code> : only change pre outer space if it contains shrink	75
<code>\microtypecontext</code> : made robust (reported by <i>Ste- phan Hennig</i>)	92		
<code>\MT@begin@catcodes</code> : fix: don't disable <code>\KV@sp@def</code>	80		
<code>\MT@define@set@key@font</code> : font: single asterisk means normal font	97		

Version 2.3a (2008/02/29)

General: error messages if pdfTeX is too old for exten- sions	125	<code>\MT@fix@catcode</code> : fix catcodes earlier, and also for the letterspace package	34
fix test for <code>soul</code> under plain TeX	52	<code>\MT@getkey</code> : fix: <code>key=val</code> in class options list	120
Documentation: add hint about babel having to be loaded first	27	<code>\MT@maybe@etex</code> : removed	41
add table of available and enabled features	7	<code>\MT@set@codes</code> : generalised	58
mention <code>soulutf8</code>	29	<code>\MT@setupfont@hook</code> : restore percent character if Mexican babel is loaded	50
Protrusion: adjust LMR quotation marks again ...	142		

Version 2.3b (2008/06/04)

General: compatibility with CJKutf8: also check for its definition	89	known slots	79
<code>\MT@exp@gcs</code> : new macro: reduce save stack size ..	41	<code>\MT@requires@latex</code> : new macro	37
<code>\MT@font@copy</code> : enable font copies also with protru- sion contexts (reported by <i>Nathan Rosenblum</i>) ..	54	<code>\MT@set@tr@codes</code> : fix: protrusion adjustment only for new fonts (reported by <i>Wolfram Schaalo</i>) ..	71
<code>\MT@get@size</code> : grouping	96	<code>\MT@tr@outer@l</code> : fix: only in horizontal mode	75
<code>\MT@noligatures@</code> : fix: warning messages for un- known slots	79	make <code>\spaceskip-aware</code> (ragged2e)	75
		<code>\MT@tr@outer@r@</code> : additional test for horizontal mode ..	76

Version 2.3c (2008/11/11)

General: luaTeX supported by default	39	coding (reported by <i>Vasile Gaburici</i>)	133
Documentation: add hint about partial incompatibil- ity with CJK	27	<code>\MT@detokenize@c</code> : fix: remove last space only (repor- ted by <i>Ulrich Durr</i>)	43
add hint about spacing being experimental	26	<code>\MT@tr@outer@r@</code> : additional test for horizontal mode (reported by <i>Sveinung Heggen</i>)	76
Inheritance: add <code>\textcommabelow[STst]</code> to QX en-			

Version 2.3d (2009/03/27)

General: default step: 1 for pdfTeX versions ≥ 1.40 ..	122	<code>\microtypesetup</code> : select font after setup	118
fix pinyin compatibility check (reported by <i>Silas S. Brown</i>)	52	<code>\MT@check@active@set</code> : warning for missing default sets	117
move setup to the very end (for <i>Colin Rourke</i>) ..	127	<code>\MT@lua</code> : update for luaTeX 0.36	41
<code>\ifMT@inannot</code> : use <code>pdfTeXcmds</code> for debugging	36	<code>\MT@set@tr@codes</code> : allow zero tracking	70
<code>\lsstyle</code> : disable for luaTeX	72	<code>\MT@set@tr@zero</code> : fix: allow switching off tracking (reported by <i>Ulrich Durr</i>)	73
make invalid in math mode	72		

<code>\MT@tr@outer@r@</code> : don't use <code>\x</code> (reported by <i>Ulrich Dirr</i>)	76
fix: don't adjust in math mode (reported by <i>Christoph Bier</i>)	75
fix: don't adjust inside discretionary (reported by <i>Maverick Woo</i>)	76
<code>\MT@tr@set@okern</code> : allow empty value for outer kerning	78
<code>\textls</code> : make math mode aware	77

B Index

Links (in blue) refer to the page where the corresponding entry is described (bold face) resp. occurs. Plain numbers (in black) refer to the code line where the corresponding entry is defined (underlined) resp. used.

Options:

<code>DVIoutput</code>	9
<code>activate</code>	6
<code>auto</code>	8
<code>babel</code>	9
<code>config</code>	9
<code>draft</code>	9
<code>expansion</code>	6
<code>factor</code>	7
<code>final</code>	9
<code>kerning</code>	7
<code>letterspace</code>	9
<code>protrusion</code>	6
<code>selected</code>	8
<code>shrink</code>	8
<code>spacing</code>	7
<code>step</code>	8
<code>stretch</code>	8
<code>tracking</code>	7
<code>unit</code>	8
<code>verbose</code>	9
<code>\add@accent</code>	2374
<code>ae</code> (package)	21 , 129
<code>amssymb</code> (package)	166
<code>article</code> (package)	23
<code>auto</code> (option)	8 , 114

B

<code>babel</code> (option)	9 , 23 , 27 , 30 , 114 , 127 , 186
<code>babel</code> (package)	3 , 5 , 17 , 23 , 24 , 27 , 50 , 80 , 100 , 126 , 127 , 130 , 183 , 184 , 186–188

C

<code>chemsym</code> (package)	183
<code>chmath</code> (package)	21 , 130 , 187
<code>CJK</code> (package)	27 , 52 , 89 , 187 , 188
<code>CJKutf8</code> (package)	89 , 188
<code>cm-super</code> (package)	8
<code>color</code> (package)	9 , 121
<code>combine</code> (package)	127 , 186
<code>config</code> (option)	9 , 21 , 31 , 116 , 184
<code>contour</code> (package)	121
<code>copyfonts</code> (option)	54 , 114 , 187
<code>crop</code> (package)	121
<code>csquotes</code> (package)	29 , 50 , 140 , 184
<code>\curr@fontshape</code>	2321 , 2323 , 2325 , 2332 , 2372

D

<code>\DeclareCharacterInheritance</code>	20 , 41 , 3093
<code>\DeclareMicrotypeAlias</code>	22 , 43 , 70 , 2750
<code>\DeclareMicrotypeBabelHook</code>	23 , 45 , 64 , 2803
<code>\DeclareMicrotypeSet</code>	10 , 32 , 57 , 59 , 71 , 2481
<code>\DeclareMicrotypeSet*</code>	10 , 2481
<code>\DeclareMicrotypeSetDefault</code>	12 , 34 , 61 , 2709
<code>\DeclareMicrotypeVariants</code>	21 , 42 , 58 , 63 , 2737
<code>\DeclareMicrotypeVariants*</code>	21
<code>deferssetup</code> (option)	50 , 114 , 118 , 185
<code>\define@newfont</code>	2314 , 2322 , 2326 , 2334
<code>\DisableLigatures</code>	25 , 40 , 62 , 2783
<code>\do@subst@correction</code>	2371
<code>docstrip</code> (package)	33
<code>draft</code> (option)	9 , 113 , 115 , 186
<code>dsfont</code> (package)	58
<code>DVIoutput</code> (option)	9 , 114 , 183

E

<code>eco</code> (package)	21 , 129
<code>\efcode</code>	1298 , 1322 , 1323 , 1363 , 1365
<code>eplain</code> (package)	25 , 30 , 33 , 37 , 49 , 120 , 188
<code>eucal</code> (package)	171
<code>eufrak</code> (package)	173

A

<code>a0poster</code> (package)	96
<code>activate</code> (option)	6 , 113 , 186

- euler (package) 87, 170, 186
eulervm (package) 21, 129, 170, 186
euroitc (package) 174
europ (package) 174
eurosans (package) 174
expansion (option) 6, 7, 11, 113, 121
- F**
- \f@family 2328, 2329
\f@size 2321, 2323, 2325, 2332, 2372
factor (option) 7, 13, 32, 115, 116, 183
fancyvrb (package) 27
final (option) 9, 32, 113, 115, 183
fix-cm (package) 55
\font 801, 1659
\font@name ... 110, 837, 839, 841, 843, 853, 854,
1548, 1553, 1587, 1594, 1604, 1607, 1646,
1659, 1671, 1673, 1678, 1680, 1683, 1686,
1689, 1721, 1776, 1893, 1895, 2314, 2322,
2331, 2334, 2355, 2357, 2359, 2373, 2385, 2405
\fontcharwd 1062
\fontdimen 991,
998, 1083, 1088, 1187, 1190, 1565, 1566,
1594, 1671, 1714, 1715, 1722, 1723, 1729,
1733, 1740, 1750, 1754, 1757, 1776, 1906, 1909
fontinst (package) 177
fontinstallationguide (package) 156
french (package) 184
frenchpro (package) 86, 183, 187
- G**
- german (package) 34, 182
graphics (package) 9, 121
- H**
- hfoldsty (package) 21, 129
hyperref (package) . 9, 50, 51, 89, 121, 184, 185, 188
- I**
- IEEEtran (package) 89
\iffontchar 1093
\ifMT@auto 240, 3758, 3805
\ifMT@babel 240, 3958, 4001
\ifMT@do 873, 936, 1525, 1925
\ifMT@document 282, 3209
\ifMT@draft 240, 3694
\ifMT@expansion 240, 3725, 3802
\ifMT@if@ 239,
704, 710, 728, 759, 2399, 3312, 3978, 3981
\ifMT@inannot 98
\ifMT@inlist@ 534, 579, 638,
661, 816, 903, 920, 929, 947, 965, 1523,
1976, 1987, 2036, 2084, 2393, 2455, 2467, 2768
\ifMT@kerning 240, 3868, 3917, 3971
\ifMT@noligatures 240, 3927
\ifMT@nonselected 1276, 2097, 2105
\ifMT@noreset 2159, 2166
\ifMT@opt@auto 3319, 3761
\ifMT@opt@DVI 3319, 3710
\ifMT@opt@expansion 3319, 3705
\ifMT@protrusion 240, 667, 689, 3713, 3851
\ifMT@selected 240, 3788, 3808
\ifMT@spacing 240, 3857, 3911, 4000
- \ifMT@tracking 240, 3846, 3905
ifpdf (package) 9, 89, 121
\iftracingmicrotypeinpdfall 104
inputenc (package)
.... 14, 27, 31, 52, 63, 84–87, 184, 186–188
- J**
- jurabib (package) 89
- K**
- kerning (option) 7, 11, 30, 113
keyval (package) 39, 59, 130, 187
\knacode 1478, 1479, 1489, 1492, 1498
\knbcodes 1471, 1472, 1488, 1491, 1497
\knbscode 1392, 1393, 1416, 1420, 1426
- L**
- ledmac (package) 29, 31, 49, 89, 184
ledpar (package) 29
\leftmargin kern 675
letterspace (option) . 9, 18, 24, 30, 43, 70, 115, 186
letterspace (package) 1,
24, 25, 30, 33, 37, 49, 70, 71, 73, 186–188
\letterspacefont 1553
lineno (package) 49
listings (package) 27, 51, 52, 63, 80, 185–187
lmodern (package) 8, 142
\LoadMicrotypeFile 22, 44, 73, 2764, 3699
\lcode 1004, 1036, 1037, 1223, 1226, 1578
\lslig 24, 54, 1664, 1667
\lstyle 24, 51, 751, 774, 779,
1647, 1836, 1839, 1844, 1847, 1858, 1862, 3702
- M**
- marvosym (package) 21, 31, 33, 135, 174, 186
mathdesign (package) 21, 130, 185
\mbx 129
memoir (package) 23, 89
\Microtype@Hook 117, 3551
\microtypecontext
.. 22, 47, 749, 2425, 2435, 3701, 3941, 3944
microtypecontext (environment) 22
\microtypesetup 10, 46, 3555, 3700
miniltx (package) 25, 30, 33, 37, 49, 188
minimal (package) 52
\MT@font 331, 819,
835, 993, 1084, 1097, 1536, 2039, 2110, 2360
\MT@abbr@ex 614
\MT@abbr@ex@c 614
\MT@abbr@ex@inh 614
\MT@abbr@kn 614
\MT@abbr@kn@c 614
\MT@abbr@kn@inh 614
\MT@abbr@nl 614
\MT@abbr@pr 614
\MT@abbr@pr@c 614
\MT@abbr@pr@inh 614
\MT@abbr@sp 614
\MT@abbr@sp@c 614
\MT@abbr@sp@inh 614
\MT@abbr@tr 614
\MT@abbr@tr@c 614

- \MT@active@features 845, 2387, 2390, 2402, 2409, 2444, 2454, 2790, 3714, 3803, 3848, 3859, 3870, 3933, 3934
 \MT@addto@annot 98
 \MT@addto@setup 653, 654, 713, 1200, 2308, 2425, 2426, 2988, 3006, 3423, 3556, 3693, 3844, 3904, 3926, 3932, 3957, 3990
 \MT@auto 1285, 3757, 3771, 3781, 3785
 \MT@auto@ 1285, 1294, 1336
 \MT@autofalse 242, 3769, 3780
 \MT@autotruer 242, 3513
 \MT@babelfalse 249
 \MT@babeltrue 249
 \MT@begin@catcodes 1977, 1978, 2018, 2772, 2785, 2810, 2824, 2846, 2865, 2879, 3098, 3526, 3527
 \MT@cat 1196, 2300, 2302
 \MT@cfg@catcodes 761, 1218, 2003, 2020
 \MT@char 1026, 1036, 1037, 1042, 1043, 1046, 1048, 1062, 1063, 1093, 1223–1227, 1322, 1323, 1325, 1326, 1363–1365, 1392, 1393, 1399, 1400, 1406, 1407, 1410, 1411, 1416–1421, 1471, 1472, 1478, 1479, 1482, 1483, 1488–1492, 1936, 2139, 2155, 2156, 2161, 2232, 2235, 2237, 2263, 2266, 3158, 3159, 3164, 3165
 \MT@char@ 2139, 2145, 2150, 2155, 2169, 2171, 2177, 2178, 2180, 2193, 2194, 2197, 2198, 2201, 2202, 2206, 2208, 2236, 2249, 2252, 2255, 2274
 \MT@charstring 2152, 2235, 2240
 \MT@check@active@set 3537, 3720, 3821, 3850, 3862, 3874
 \MT@check@font 815, 2384, 2446
 \MT@check@font@cx 2388, 2446
 \MT@check@range 3283, 3285
 \MT@check@range@ 3285, 3286
 \MT@check@rlist 3236, 3276
 \MT@check@rlist@ 3276, 3277
 \MT@check@step 3809
 \MT@checklist@ 881, 896, 1922
 \MT@checklist@family 914
 \MT@checklist@font 958
 \MT@checklist@size 942
 \MT@checksetup 3565, 3576, 3583, 3602, 3622
 \MT@clear@options 192, 216, 231, 649, 3687
 \MT@clist@break 506, 909, 937, 953, 967
 \MT@clist@function 506
 \MT@cnt@encoding 3175, 3183, 3184
 \MT@cnt@family 3181, 3190, 3191
 \MT@cnt@series 3188, 3197, 3198
 \MT@cnt@shape 3195, 3203, 3204
 \MT@config@file 3516, 3524, 3525, 3529, 3530, 3533
 \MT@context 2060, 2072, 2091, 2118
 \MT@copy@font 804, 832, 2984, 2990, 3002, 3009, 3440, 3442, 3711
 \MT@copy@font@ .. 832, 2984, 2990, 3002, 3009, 3440
 \MT@count 444, 445, 598, 1055, 1062, 1064, 1065, 1070, 1075, 1076, 1080, 1088, 1119, 1181
 \MT@curr@file 1981, 1982, 1991, 1992, 2774, 2775, 2957, 3120, 3529, 3992
 \MT@curr@list@name ... 1086, 1122, 1212, 1332, 1933, 1962, 2129, 2269, 2277, 2282, 2290, 2296
 \MT@curr@ls 1584, 1668, 1671
 \MT@curr@set@name 2506, 2508–2511, 2513, 2514, 2519, 2524, 2528, 2529, 2554, 2558, 2587, 2617, 2623, 2629, 2633, 2634, 2861, 2919, 2937, 2952, 2964, 3028, 3032, 3037, 3040, 3042, 3046, 3049, 3052, 3062, 3075, 3081, 3089, 3121, 3123
 \MT@declare@char@inh 3103, 3108, 3118
 \MT@declare@sets 2489, 2495, 2507, 2792
 \MT@DeclareSet 2484, 2486, 2503
 \MT@DeclareSetAndUseIt 2483, 2502
 \MT@DeclareVariants 2740, 2741, 2743
 \MT@def@bool@opt 3394, 3407–3409, 3418, 3430, 3438, 3447, 3454
 \MT@def@n 320, 2477, 2478
 \MT@default@ex@set 2725
 \MT@default@kn@set 2725
 \MT@default@pr@set 2725
 \MT@default@sp@set 2725
 \MT@default@tr@set 2725
 \MT@define@code@key . 2901, 2944–2947, 3143–3145
 \MT@define@code@key@font 2924, 2949, 3147
 \MT@define@code@key@size 2912, 2948, 3146
 \MT@define@opt@key .. 2950, 2966–2969, 3085–3087
 \MT@define@optionX 3555, 3611, 3612
 \MT@define@optionX@ 3613, 3641, 3643, 3644
 \MT@define@set@key@ 2517, 2673–2676
 \MT@define@set@key@font 2621, 2678
 \MT@define@set@key@size 2547, 2677
 \MT@detokenize@c 361, 2147, 2233
 \MT@detokenize@n 361, 2265
 \MT@dimen@six 990, 1056, 1080, 1119, 1749, 1752, 1902
 \MT@dinfo 86
 \MT@dinfo@list 894, 904, 907, 912, 921, 924, 930, 932, 940, 948, 951, 956, 966, 969, 973
 \MT@dinfo@nl 86
 \MT@DisableLigatures 2783
 \MT@do@font 596, 1006, 1298, 1429, 1499, 1578
 \MT@documentfalse 282
 \MT@documenttrue 282, 3936
 \MT@dofalse 873, 885, 908, 925, 952, 970
 \MT@dotrue 873, 876, 905, 922, 931, 949, 1917
 \MT@draftfalse 245, 3456
 \MT@drafttrue 245, 3458
 \MT@edef@n 322, 2096, 2122, 2471, 2907, 2957, 2959, 3119, 3134, 3263, 3487
 \MT@encoding 863, 962, 2055, 2067, 2147, 2233, 2264, 2281, 2288, 2296
 \MT@encoding@check 3132, 3136
 \MT@end@catcodes 1979, 2022, 2780, 2794, 2821, 2843, 2862, 2876, 2890, 3113, 3528
 \MT@error 75, 221, 641, 716, 1837, 1845, 1954, 1967, 2585, 2704, 2731, 2798, 2976, 2993, 3018, 3432, 3449, 3588, 3606, 3630, 3762, 3774, 3838, 3906, 3912, 3918, 3953
 \MT@ex@cc@name 1269, 1271, 1338, 1339, 1349, 1355, 1356, 1368, 2827
 \MT@ex@context 835, 2438, 2476
 \MT@ex@doc@contexts 2476
 \MT@ex@factor 250, 1286, 1342
 \MT@ex@factor@ . 1286, 1300, 1303, 1312, 1313, 1336
 \MT@ex@inh@name 1324–1326
 \MT@ex@level 250, 3804, 3806
 \MT@ex@max 262, 1315, 1316

- \MT@ex@min 262, 1318, 1319
 \MT@ex@setname 2696
 \MT@ex@split@val 1310
 \MT@exp@cs 317, 320, 323,
 326, 329, 846, 848, 857, 946, 1047, 1106,
 1109, 1139, 1143, 1153, 1156, 1159, 1326,
 1411, 1483, 2083, 2403, 2410, 2411, 2415,
 2457, 2468, 2553, 2918, 2935, 3165, 3238, 3281
 \MT@exp@gcs 317, 321, 325, 327, 330
 \MT@exp@one@n 332,
 366, 513, 637, 807, 809, 858, 900, 963,
 1522, 1986, 2384, 2391, 2421, 2454, 2465,
 2476, 2495, 2672, 2689, 2718, 2767, 3108, 3941
 \MT@exp@two@c 334,
 352, 358, 367, 548, 806, 839, 841, 843,
 852, 1646, 1680, 1683, 1686, 2146, 2151, 2234
 \MT@exp@two@n 336, 918, 927, 2523, 2628
 \MT@expandfont 1273, 1289, 1293
 \MT@expansion 821, 1261, 3831
 \MT@expansionfalse 241, 3706, 3798
 \MT@expansiontrue 241, 3512
 \MT@extra@context 2476, 2813, 2827,
 2850, 2868, 2882, 2930, 2933, 2934, 2936,
 2970, 2985, 2991, 3003, 3008, 3094, 3235,
 3239, 3242, 3245, 3248, 3252, 3253, 3255, 3282
 \MT@extra@inputenc 3093, 3117, 3122, 3123
 \MT@factor@default 274, 3493, 3717
 \MT@family ... 807, 863, 919, 962, 2056, 2759, 2760
 \MT@familyalias
 ... 808, 809, 869, 926, 928, 2066, 2068, 2761
 \MT@feat 873, 995, 1028, 1101, 1103,
 1105, 1106, 1108, 1109, 1115, 1117, 1120,
 1128–1132, 1134, 1136–1140, 1143, 1144,
 1149, 1153, 1156, 1159, 1162–1164, 1173,
 1174, 1179, 1197, 1649, 1952, 1954, 1957,
 1958, 1962, 1964, 1967, 2092, 2094, 2096,
 2100, 2104, 2108, 2109, 2112, 2120, 2122,
 2123, 2125, 2132, 2136, 2270, 2271, 2300, 2302
 \MT@features
 634, 2476, 2489, 2672, 2683, 2712, 2954, 3103
 \MT@features@long 634, 637, 642, 2451, 3115
 \MT@file@list 1973,
 1975, 1983, 1986, 1988, 1993, 1996, 2767, 2771
 \MT@find@file 807, 809, 1973
 \MT@fix@catcode 5
 \MT@fix@font@set 2542, 3712
 \MT@font 331,
 806, 852, 859, 991, 998, 1004, 1005, 1036,
 1037, 1042, 1043, 1062, 1063, 1083, 1088,
 1093, 1187, 1190, 1223, 1224, 1226, 1227,
 1294, 1298, 1322, 1323, 1363, 1365, 1392,
 1393, 1399, 1400, 1406, 1407, 1416–1418,
 1420, 1421, 1426–1428, 1471, 1472, 1478,
 1479, 1488, 1489, 1491, 1492, 1497, 1498,
 1522, 1526, 1583, 1926, 2304, 2355–2357,
 2359, 2372, 2373, 2384, 2386, 2391, 2413, 2422
 \MT@font@copy 835, 836, 843, 844, 852–854
 \MT@font@list 2304, 2384–2386, 2445
 \MT@font@orig 836
 \MT@font@sets 2527, 2542, 2632, 3712
 \MT@gdef@n 320, 2806, 2820,
 2842, 2875, 2889, 3042, 3062, 3075, 3125, 3545
 \MT@get@axis 2646–2649, 2659
 \MT@get@basefamily 1985, 2023
 \MT@get@basefamily@ 2029, 2032
 \MT@get@char@unit 1027, 1151, 1177, 1265
 \MT@get@charwd 1061, 1080, 1154, 1177
 \MT@get@config 3516
 \MT@get@ex@opt 1264, 1280, 1336
 \MT@get@ex@opt@ 1344–1346, 1348, 1354
 \MT@get@font 2627, 2637
 \MT@get@font@ 2638, 2643, 2942
 \MT@get@font@and@size 2929, 2941
 \MT@get@font@dimen 1082, 1157
 \MT@get@font@dimen@six . 978, 990, 1376, 1455, 1537
 \MT@get@highlevel ... 2522, 2532, 2661, 2906, 3133
 \MT@get@inh@list 981, 1267, 1379, 1458, 2118
 \MT@get@listname 2038, 2094, 2120
 \MT@get@listname@ 2038
 \MT@get@ls@basefont 1552, 1670, 1677
 \MT@get@opt 979, 1126, 1377, 1456
 \MT@get@range 2551, 2561, 2916
 \MT@get@size 2568, 2573, 2582, 2601, 2657
 \MT@get@slot 1025, 1935, 2139, 3157, 3163
 \MT@get@space@unit
 ... 1151, 1251, 1390, 1397, 1404, 1469, 1476
 \MT@get@tr@opt 1539, 1617
 \MT@get@tr@opt@ 1635–1638, 1640
 \MT@get@unit 1159, 1167, 1628
 \MT@get@unit@ 1167
 \MT@getkey 3674, 3689
 \MT@glet 316, 327, 839, 1680,
 1709, 1839, 1847, 2353, 2446–2449, 2761,
 2984, 3002, 3178, 3440, 3442, 3888, 3893, 3995
 \MT@glet@nc 326, 1721, 1893, 2132,
 2373, 2445, 2511, 2513, 2519, 2623, 2835,
 3028, 3032, 3037, 3040, 3046, 3049, 3052, 3081
 \MT@glet@nn 329, 2464, 3542
 \MT@gobble@settings 989, 996
 \MT@if@false 239,
 700, 707, 724, 741, 2396, 3280, 3972, 3979
 \MT@if@list@exists
 ... 977, 1263, 1279, 1375, 1454, 1538, 2091
 \MT@if@true
 239, 701–703, 708, 709, 725–727, 754, 758,
 2389, 3289, 3293, 3301, 3306, 3973–3977, 3980
 \MT@ifdefined@c@T
 ... 340, 704, 730, 1045, 1069, 1324, 1409,
 1481, 1571, 2221, 2759, 3122, 3225, 3551, 3554
 \MT@ifdefined@c@TF
 ... 340, 668, 1542, 1555, 1558, 1569,
 1668, 1746, 1899, 1930, 1931, 2095, 2121, 3234
 \MT@ifdefined@n@T
 ... 340, 658, 897, 915, 943, 959, 1046,
 1162, 1198, 1325, 1349, 1410, 1482, 1620,
 1622, 1641, 2074, 2299, 2509, 2754, 2830,
 2960, 3184, 3191, 3198, 3204, 3235, 3245, 3674
 \MT@ifdefined@n@TF 340, 869,
 879, 898, 916, 944, 960, 1128, 1136, 1338,
 1355, 1920, 1957, 2062, 2147, 2698, 2701,
 2727, 3211, 3262, 3271, 3335, 3538, 3541, 3939
 \MT@ifdim .. 453, 568, 569, 573, 574, 2584, 2593,
 3287–3289, 3291, 3292, 3299–3301, 3304, 3305
 \MT@ifdimen 418, 2612, 3031, 3039, 3051, 3502

- `\MT@ifempty` . 373, 1004, 1005, 1034, 1040, 1240, 1241, 1389, 1396, 1403, 1426–1428, 1440–1442, 1444–1446, 1468, 1475, 1497, 1498, 1514, 1515, 1727, 1728, 1738, 1756, 1870, 1891, 1892, 1904, 2456, 2488, 2493, 2534, 2540, 2562, 2563, 2576, 2577, 2651, 2662, 2682, 2687, 2711, 2716, 2789, 2857, 2951, 2956, 2970, 2983, 2991, 3001, 3008, 3060, 3102, 3107, 3137, 3348, 3379, 3518, 3562, 3619
`\MT@ifint` 386, 2200, 2858, 3061, 3486, 3495
`\MT@ifstreq` 460, 1204, 1953, 2093, 2108, 2250, 2253, 2420, 2626, 2928, 3030, 3048, 3050, 3226, 3351, 3353, 3356, 3359, 3381, 3383, 3397, 3398, 3464, 3465, 3470, 3474, 3501, 3564, 3571, 3575, 3582, 3621, 3627, 4003
`\MT@ineclist` 534, 637, 659, 900, 918, 927, 963, 1522, 1975, 1986, 2384, 2391, 2454, 2767
`\MT@in@rlist` 562, 946, 2083
`\MT@in@rlist@` 562
`\MT@in@rlist@` 562
`\MT@in@tlist` 550, 2035, 2465
`\MT@in@tlist@` 550
`\MT@inannotfalse` 99
`\MT@inannottrue` 99
`\MT@increment` 598, 3183, 3190, 3197, 3203
`\MT@info` 75, 90, 95, 3471, 3659, 3663
`\MT@info@missing@char` 1065, 1091
`\MT@info@nl` 75, 91, 92, 96, 669, 1092, 3462, 3472, 3525, 3539, 3543, 3709, 3711, 3716, 3723, 3805, 3832, 3849, 3855, 3861, 3866, 3873, 3878, 3909, 3915, 3921, 3949, 3960
`\MT@inh@do` 2131, 3148
`\MT@inh@feat` 3093, 3102, 3105, 3116
`\MT@inh@split` 3150, 3154
`\MT@inlist@false` 534, 538, 551, 563, 2399
`\MT@inlist@true` 534, 540, 558, 570, 575, 2399
`\MT@is@active` 2146, 2213
`\MT@is@char` 2151, 2234, 2240
`\MT@is@composite` 2149, 2261
`\MT@is@feature` 636, 2494, 2688, 2717
`\MT@is@letter` 2144, 2167, 2237, 2266
`\MT@is@number` 2183, 2188
`\MT@is@symbol` 2148, 2231
`\MT@iterate` 584
`\MT@kerning` 824, 1451, 3877
`\MT@kerningfalse` 247
`\MT@kerningtrue` 247
`\MT@kn@c@name` 1460, 1462, 1505, 2882
`\MT@kn@context` 2439, 2476
`\MT@kn@doc@contexts` 2476
`\MT@kn@factor` 250
`\MT@kn@factor@` 1128
`\MT@kn@inh@name` 1481–1483
`\MT@kn@max` 262
`\MT@kn@min` 262
`\MT@kn@setname` 2696
`\MT@kn@split@val` 1466
`\MT@kn@unit` 256
`\MT@kn@unit@` 1136, 1508
`\MT@led@kern` 665
`\MT@led@unhbox@line` 665
`\MT@ledmac@setup` 665, 722
`\MT@let@cn` 328, 329, 330, 870, 985, 1271, 1339, 1383, 1462, 1621, 1623, 1627, 1952, 1964, 2088, 2092, 2125, 2355, 2605, 2964, 3121
`\MT@let@enc` 326, 664, 1164, 2104, 2136, 2394, 3264, 3269, 3357, 3360, 3578, 3585, 3840
`\MT@let@nn` 329, 1129, 1134, 1137, 1149, 1356, 1359, 1642
`\MT@letterspace` 278, 1542, 1621, 1631, 3896, 3897, 3899
`\MT@letterspace@` 1013, 1069, 1070, 1542, 1543, 1546, 1549, 1553, 1557, 1565, 1577–1579, 1584, 1594, 1752, 1756, 1856, 1871–1873, 1905, 1908
`\MT@letterspace@default` 278, 3897
`\MT@listname` 2038, 2085, 2088, 2095, 2096, 2098, 2100, 2121, 2122, 2124, 2125, 2129, 2132, 3165, 3170
`\MT@load@inputenc` 1205, 1208, 1217
`\MT@load@list` 983, 1269, 1381, 1460, 1950
`\MT@loop` 584, 592, 3268
`\MT@lower` 2555, 2561, 2919
`\MT@ls@adjust` 1592, 1867
`\MT@ls@adjust@` 1852, 1874
`\MT@ls@adjust@empty` 1853, 1867
`\MT@ls@adjust@relax` 1854, 1867
`\MT@ls@basefont` 1677, 1686, 1689, 1690
`\MT@ls@outer@k` 1595, 1599, 1605, 1769, 1783, 1825, 1913
`\MT@ls@set@ls` 1858, 1861, 1867
`\MT@ls@too@large` 1873, 1876, 3899
`\MT@lsfont` 1548, 1553, 1565, 1566, 1575, 1578, 1579, 1583, 1646, 1655, 1699, 1701, 1714, 1715, 1722, 1723, 1729, 1733, 1740, 1750, 1754, 1757, 1906, 1909
`\MT@lslig` 1667
`\MT@lua` 307, 390, 422, 464
`\MT@map@clist@` 506
`\MT@map@clist@c` 506, 845, 1934, 2390, 2402, 2409, 2444, 2451, 2489, 2491, 2683, 2685, 2712, 2714, 2954, 3103, 3105, 3115, 3672
`\MT@map@clist@n` 506, 878, 1919, 2476, 2520, 2549, 2624, 2672, 2744, 2804, 2904, 2914, 2926, 3044, 3058, 3160, 3343, 3346, 3375, 3377, 3406, 3481, 3560, 3617
`\MT@map@tlist@` 524
`\MT@map@tlist@c` 524, 553, 564, 848, 1047, 1326, 1411, 1483, 2042, 2415, 3236, 3281, 3712
`\MT@map@tlist@n` 524, 2014, 2242, 3177
`\MT@maybe@do` 873, 975, 1261, 1373, 1452, 1524
`\MT@maybe@etex` 283
`\MT@maybe@rem@from@list` 2415, 2419
`\MT@MT` 2, 75, 78, 81, 83, 85, 90, 91, 93, 94, 204, 212, 223, 648, 717, 719, 2993, 3519, 3659, 3663, 3710, 3953, 3954, 4007
`\MT@next@listname` 2045, 2054
`\MT@next@listname@` 2063, 2075, 2081
`\MT@nl@ligatures` 1926, 2783
`\MT@nl@setname` 1918, 2783
`\MT@noligatures` 826, 1915, 3928
`\MT@noligatures@` 1699, 1701, 1926, 1929
`\MT@noligaturesfalse` 244
`\MT@noligaturestrue` 244, 2791
`\MT@nonselectedfalse` 1276, 1290

\MT@nonselectedtrue	1276, 1278	\MT@preset@sp	1433
\MT@noestfalse 2166, 2173, 2182, 2203, 2250, 2253		\MT@preset@sp@	1433
\MT@noesttrue	2142, 2166, 2191	\MT@ProcessOptionsWithKV	3667, 3690
\MT@old@cmd	65, 70–73	\MT@protrusion	820, 975, 3722
\MT@opt@autofalse	3320	\MT@protrusionfalse	240
\MT@opt@autotrue	3320, 3408	\MT@protrusiontrue	240, 3509
\MT@opt@def@set	3334, 3362, 3387	\MT@rbba@expansion	629
\MT@opt@DVIfalse	3321	\MT@rbba@kerning	629
\MT@opt@DVITrue	3321, 3411, 3414	\MT@rbba@protrusion	629
\MT@opt@expansionfalse	3319	\MT@rbba@spacing	629
\MT@opt@expansiontrue	3319	\MT@rbba@tracking	629
\MT@options	2892	\MT@register@font	829, 2386, 2447
\MT@optwarn@admissible	3322, 3399, 3474	\MT@register@font@cx	2408, 2447
\MT@optwarn@nan	3326, 3488, 3497	\MT@register@subst@font	2361, 2385, 2448
\MT@orig@add@accent	2374	\MT@register@subst@font@cx	2401, 2448
\MT@orig@foreign@language	3966, 3968	\MT@rem@from@clist	545, 858, 2421, 2896
\MT@orig@pickupfont	747, 788, 793, 2309, 2376	\MT@rem@from@list	848, 856
\MT@orig@py@macron	785, 789, 794	\MT@rem@last@space	361
\MT@orig@select@language	3961, 3963	\MT@repeat	584, 594, 3274
\MT@outer@kern	1594, 1597, 1603, 1605, 1606, 1611, 1671, 1672, 1674, 1825, 1887, 1913	\MT@requires@latex	144, 193, 651, 743, 772, 1645, 2307, 3327, 3480, 3667, 3671, 3994
\MT@outer@space		\MT@requires@luatex	299, 307, 388, 420, 462, 1843, 2974, 3429, 3614, 3847, 3858, 3869
....	1587, 1588, 1712, 1765, 1767, 1768, 1781, 1782, 1797, 1798, 1813, 1814, 1818, 1819	\MT@requires@pdftex	295, 386, 418, 460, 665, 803, 812, 823, 825, 833, 1011, 1053, 1067, 1301, 1372, 1451, 1519, 1696, 1915, 2783, 2972, 3074, 3080, 3427, 3511, 3613, 3732, 3759, 3784, 3835, 3925, 3998
\MT@pdf@annot	98	\MT@res@a	483, 485, 536, 543, 546, 548, 552, 557
\MT@pdftex@no	148, 198, 202, 296	\MT@res@b	484, 485, 546–548, 556, 557
\MT@permute 2819, 2841, 2855, 2874, 2888, 3128, 3174		\MT@reset@context	2429, 2432, 2436, 2463
\MT@permute@	3174	\MT@reset@context@	2436, 2463
\MT@permute@@	3174	\MT@reset@ef@codes	1266, 1288, 1301, 1351
\MT@permute@@@	3174	\MT@reset@ef@codes@	1296, 1304, 1308
\MT@permute@@@@	3202, 3207	\MT@reset@kn@codes	1457, 1464, 1494
\MT@permute@@@@@	3215, 3217	\MT@reset@kn@codes@	1494
\MT@permute@define	3208, 3218–3220, 3260	\MT@reset@pr@codes	980, 987, 1008
\MT@permute@reset	3177, 3266	\MT@reset@pr@codes@	1008
\MT@permutelist		\MT@reset@sp@codes	1378, 1385, 1423
....	2813, 2827, 2853, 2868, 2882, 2930, 2931, 2936, 3126, 3230, 3231, 3235, 3239, 3242, 3245–3248, 3255, 3256, 3282, 3313, 3314	\MT@reset@sp@codes@	1423
\MT@pickupfont	790, 795, 2370, 2378	\MT@restore@catcodes ...	5, 7, 8, 216, 231, 649, 4016
\MT@plain	124, 145	\MT@restore@p@h	698, 710, 759
\MT@pr@c@name	983, 985, 1231, 2813	\MT@saved@setupfont	3657, 3664
\MT@pr@context	835, 2438, 2476	\MT@scale	605, 1056, 1102, 1116, 1119, 1313
\MT@pr@doc@contexts	2476	\MT@scale@factor	1058, 1077, 1100, 1246, 1369
\MT@pr@factor	250, 3496, 3717, 3718	\MT@scale@to@em ..	1035, 1041, 1053, 1074, 1252, 1391, 1398, 1405, 1470, 1477, 1630, 1747, 1900
\MT@pr@factor@	1128	\MT@selectedfalse	243
\MT@pr@inh@name	1045, 1046, 1048	\MT@selectedtrue	243
\MT@pr@level	250, 3715, 3716	\MT@series	863, 962, 2057, 2069
\MT@pr@max	262	\MT@set@all@ex	1296, 1370
\MT@pr@min	262	\MT@set@all@kn	1494, 1516
\MT@pr@setname	2696	\MT@set@all@pr	1001, 1008, 1242
\MT@pr@split@val	1032	\MT@set@all@sp	1423, 1448
\MT@pr@unit	256, 3503, 3719	\MT@set@babel@context	3938, 3964, 3969, 3983
\MT@pr@unit@	1136, 1234	\MT@set@codes ..	986, 1016, 1272, 1384, 1463, 1965
\MT@preset@aux	1236, 1238, 1240, 1241, 1244, 1510, 1512, 1514, 1515	\MT@set@curr@ls	1584
\MT@preset@aux@factor 1236, 1244, 1440–1442, 1510		\MT@set@curr@ok	1597, 1598, 1611, 1802
\MT@preset@aux@space .	1238, 1244, 1444–1446, 1512	\MT@set@curr@os	1586, 1803
\MT@preset@ex	1350, 1367	\MT@set@default@set	2712, 2718, 2725
\MT@preset@kn	1503	\MT@set@ex@codes	1292, 3789, 3791
\MT@preset@kn@	1503	\MT@set@ex@codes@n	1276, 1292, 3791
\MT@preset@pr	1229	\MT@set@ex@codes@s	1262, 3789
\MT@preset@pr@	1229		

- \MT@set@ex@heirs 1326, 1362
 \MT@set@inh@list 3099, 3101
 \MT@set@inputenc 982, 1195, 1268, 1380, 1459, 2130
 \MT@set@inputenc@ 1198, 1200
 \MT@set@kn@codes 1453
 \MT@set@kn@heirs 1483, 1487
 \MT@set@listname
 984, 1127, 1270, 1337, 1382, 1461, 1619, 2269
 \MT@set@lsbasefont 1686
 \MT@set@lsfont 1582, 1645, 1655
 \MT@set@named@keys 2815, 2829, 2851, 2870, 2884, 2892
 \MT@set@pr@codes 976
 \MT@set@pr@heirs 1049, 1222
 \MT@set@sp@codes 1374
 \MT@set@sp@heirs 1411, 1415
 \MT@set@tr@codes 1534, 1650, 1660
 \MT@set@tr@zero 1544, 1686, 3641
 \MT@SetExpansion 2825, 2827
 \MT@SetExtraKerning 2880, 2882
 \MT@SetExtraSpacing 2866, 2868
 \MT@SetProtrusion 2811, 2813
 \MT@SetTracking 2847, 2849
 \MT@setup@ 650, 653, 3421, 3422, 3995
 \MT@setup@contexts 2428, 2443
 \MT@setupfont .. 802, 2364, 3657, 3660, 3664, 3698
 \MT@setupfont@hook 699, 723, 729, 734, 759, 766, 802
 \MT@shape 863, 962, 2058, 2070
 \MT@shorthandoff 3947, 3978, 3981
 \MT@show@pdfannot 105
 \MT@shrink 259, 1283,
 3729, 3730, 3734, 3735, 3738, 3742, 3794, 3807
 \MT@shrink@ 1283, 1294, 1336
 \MT@shrink@default 275
 \MT@size 569, 573, 574, 863, 962
 \MT@size@name 562, 2085
 \MT@sp@c@name 1381, 1383, 1435, 2868
 \MT@sp@context 2439, 2476, 4003
 \MT@sp@doc@contexts 2476
 \MT@sp@factor 250
 \MT@sp@factor@ 1128
 \MT@sp@inh@name 1409–1411
 \MT@sp@max 262
 \MT@sp@min 262
 \MT@sp@setname 2696
 \MT@sp@split@val 1387
 \MT@sp@unit 256
 \MT@sp@unit@ 1136, 1438
 \MT@spacing 824, 1372, 3865
 \MT@spacingfalse 246
 \MT@spacingtrue 246
 \MT@split@codes 1018, 1022
 \MT@split@name 806, 863
 \MT@step 259, 1284,
 3732, 3733, 3749, 3756, 3808, 3811, 3812, 3816
 \MT@step@ 1284, 1294, 1336
 \MT@step@default 275
 \MT@stretch 259, 1282, 3726,
 3727, 3730, 3734, 3736, 3741, 3744, 3793, 3807
 \MT@stretch@ 1282, 1294, 1336
 \MT@stretch@default 275, 3727
 \MT@temp 1003–1006, 1425–
 1429, 1496–1499, 1717, 1721, 1724, 1729,
 1732, 1737, 1794, 1797, 1804, 1807, 1810,
 1813, 1818, 1825, 1828, 1890, 1893, 1910,
 2645, 2670, 3670, 3675, 3680, 3681, 3683, 3686
 \MT@tempencoding 3210, 3211, 3221, 3227
 \MT@tempfamily 3222
 \MT@tempseries 3223
 \MT@tempshape 3224
 \MT@tempsize 3178, 3225, 3234, 3236, 3240
 \MT@test@ast 1745, 1898, 2533, 2538
 \MT@textls 49, 1853, 1854, 1856
 \MT@the@pr@code 1010, 1036, 1042, 1580
 \MT@the@pr@code@tr 1010, 1580
 \MT@tlist@break 524, 559, 581, 2047, 3316
 \MT@toks 238, 1092, 1122, 1332,
 2143, 2171, 2178, 2225, 2255, 2274, 2286, 2294
 \MT@tr@c@name
 ... 1620–1623, 1627, 1641, 1642, 2852, 2859
 \MT@tr@context 2439, 2476
 \MT@tr@doc@contexts 2476
 \MT@tr@factor@ 1629
 \MT@tr@font@list 1520
 \MT@tr@ispace 1555, 1556, 1635
 \MT@tr@ligatures 1571, 1638, 1698, 1701
 \MT@tr@max 262, 1877–1879
 \MT@tr@min 262, 1881–1883
 \MT@tr@noligatures 1571, 1695
 \MT@tr@okern 1569, 1637
 \MT@tr@ospace 1558, 1559, 1635
 \MT@tr@outer@icr 1807, 1830
 \MT@tr@outer@icr@ 1830
 \MT@tr@outer@l 1590, 1761
 \MT@tr@outer@next 1789, 1832
 \MT@tr@outer@r 1613, 1789, 1830
 \MT@tr@outer@re 1789
 \MT@tr@set@okern 1570, 1887
 \MT@tr@set@okern@ 1891, 1892, 1897
 \MT@tr@set@space 1561, 1713
 \MT@tr@set@space@ 1718–1720, 1726
 \MT@tr@set@space@@ 1731, 1736, 1739, 1744
 \MT@tr@setname 2696
 \MT@tr@unit@ 1622, 1746, 1899
 \MT@tracking 813, 1520, 1650, 2366, 3641, 3642, 3854
 \MT@tracking@ 1520, 3642
 \MT@trackingfalse 248
 \MT@trackingtrue 248
 \MT@try@order 2042, 2050
 \MT@undefined@char 2218, 2220, 2230
 \MT@upper 2555, 2561, 2919
 \MT@use@set 2683, 2689, 2696
 \MT@val 2456–2458, 2460, 2466, 2468,
 2471, 2521, 2525, 2533, 2535, 2550–2552,
 2564, 2567, 2569, 2572, 2574, 2575, 2581,
 2583, 2584, 2586, 2589, 2591, 2602, 2605,
 2609, 2612–2614, 2616, 2618, 2625–2627,
 2630, 2639–2641, 2653, 2655, 2660, 2662,
 2664, 2666, 2905, 2907, 2915–2917, 2927–
 2929, 2932, 2937, 3131, 3132, 3134, 3138,
 3159, 3165, 3169, 3170, 3335, 3336, 3339,
 3347, 3348, 3351, 3353, 3356, 3359, 3378,
 3379, 3381, 3383, 3561, 3562, 3564, 3571,
 3575, 3582, 3588, 3618, 3619, 3621, 3627, 3630
 \MT@variants 2035, 2737, 2747
 \MT@vinfo 75, 92, 819, 891, 1131,
 1140, 1144, 1179, 1340, 1357, 1536, 1546,

- 1937, 1946, 1958, 1982, 1992, 1995, 2098,
2100, 2106, 2437, 2460, 2769, 2775, 3462,
3469, 3567, 3573, 3579, 3586, 3624, 3629, 3940
\MT@warn@ascii 2180, 2273
\MT@warn@axis@empty 2652, 2663, 2668
\MT@warn@code@too@large 1106, 1109, 1113
\MT@warn@err 75, 3466, 3467
\MT@warn@ex@too@large 1316, 1319, 1330
\MT@warn@lua 3836, 3847, 3858, 3869
\MT@warn@maybe@inputenc ... 1932, 2287, 2295, 2298
\MT@warn@nodim 3023, 3033, 3041, 3053
\MT@warn@number@too@large 2207, 2279
\MT@warn@preset@tewidth ... 1235, 1255, 1439, 1509
\MT@warn@rest 2160, 2284
\MT@warn@tracking@DVI 1547, 3881
\MT@warn@unknown 2157, 2292
\MT@warning 66, 75, 93, 1878,
1882, 1940, 2109, 2510, 2616, 2669, 2755,
2777, 2860, 2961, 3024, 3064, 3077, 3139,
3228, 3246, 3313, 3466, 3471, 3551, 3649–3651
\MT@warning@nl 75,
87, 94, 201, 222, 647, 681, 690, 736, 992,
1084, 1120, 1172, 1212, 1256, 1331, 1706,
2274, 2280, 2285, 2293, 2337, 2832, 3010,
3323, 3329, 3339, 3467, 3472, 3504, 3532,
3546, 3695, 3751, 3795, 3814, 3883, 3985, 4004
\MT@while@num 590, 596, 762–764
\MT@with@babel@and@T
... 657, 701–703, 725–727, 3973–3977, 3980
\MT@with@package@T 655, 705, 708, 709,
715, 722, 732, 744, 758, 760, 773, 784, 2348
\MT@xadd 103, 492, 1983, 1988, 1993,
1996, 2403, 2411, 2468, 2553, 2771, 2918, 3165
\MT@xaddb 499, 2935, 3238
\MT@xdef@n .. 322, 2699, 2702, 2729, 2734, 2758,
2859, 2930, 2952, 3089, 3123, 3255, 3336, 3338
- N**
- \normalfont 801, 3824, 3828
\nullfont 801
- P**
- \pdfadjustinterwordglue 3643, 3860
\pdfadjustspacing 3612, 3804, 3827
\pdfappendkern 3645, 3872
\pdfcopyfont 843
pdfcpot (package) 4, 27, 29, 50, 59, 88, 182
\pdffontexpand 1294
\pdfnoligatures 1939, 1945
\pdfprependkern 3644, 3871
\pdfprotrudechars 3611, 3715, 3851, 3892
pdftexcmds (package) 36, 188
\pdftracingfonts 298
\pickup@font 747, 788, 790,
793, 795, 2336, 2338, 2347, 2370, 2376, 2378
pifont (package) 89
pinyin (package) 52, 187, 188
protrusion (option) 6, 7, 11, 113
PSNFSS (package) 177
pstricks (package) 9, 121
pxfonts (package) 21, 129
- Q**
- qfonts (package) 21, 129, 185
- R**
- ragged2e (package) 75, 188
relsize (package) 96
\rightmarginkern 674
\rpcode 1005, 1042, 1043, 1224, 1227, 1579
- S**
- selected (option) 8, 14, 32, 64, 83, 114, 183
\selectfont 1651, 2431, 2441, 3556, 3991
\set@fontsize 2609
\SetExpansion 14, 36, 2823
\SetExtraKerning 19, 38, 2878
\SetExtraSpacing 18, 39, 2864
\SetProtrusion 13, 35, 2809
\SetTracking 16, 37, 2845
\sfcode 4002
\shbrcode 1406, 1407, 1418, 1421, 1428
\showhyphens 3822, 3826
shrink (option) ... 8, 16, 32, 115, 122, 123, 185, 187
slantsc (package) 156
soul (package) 4, 29, 52, 75, 186, 188
soulutf8 (package) 29, 188
\spacefactor 1772
\spaceskip 1773, 1776
spacing (option) 7, 11, 26, 30, 113
\stbrcode 1399, 1400, 1417, 1421, 1427
step (option) . 8, 30, 32, 115, 122, 123, 183, 185, 187
stretch (option) ... 8, 16, 25, 32, 115, 121–123, 187
- T**
- \tagcode 1575, 1931, 1936
tex4ht (package) 50, 186
\textls 24,
52, 53, 752, 775, 780, 1593, 1648, 1663, 1852
\textls* 24
\textmicrotypecontext 22, 48, 748, 2435
tipa (package) 89
trace (package) 90, 186
\tracingmicrotype 86
\tracingmicrotypeinpdf 97
\tracingmicrotypeinpdfallfalse 104
\tracingmicrotypeinpdfalltrue 104
tracking (option) 7, 11, 24, 30, 113
txfonts (package) 21, 129
- U**
- ucs (package) 86, 188
ulgothic (package) 21
unit (option) 8, 31, 184–186
\UseMicrotypeSet 11, 33, 60, 72, 2504, 2680
- V**
- verbose (option) 9, 30, 32, 115, 183, 184, 188
- W**
- wordcount (package) 48, 187
- X**
- \xspaceskip 1775, 1778
- Y**
- yfonts (package) 24
- Z**
- zefonts (package) 21, 129

C The L^AT_EX Project Public License

LPPL Version 1.3c 2006-05-20

Copyright 1999, 2002–2006 L^AT_EX3 Project

Everyone is allowed to distribute verbatim copies of this license document, but modification of it is not allowed.

Preamble

The L^AT_EX Project Public License (LPPL) is the primary license under which the the L^AT_EX kernel and the base L^AT_EX packages are distributed.

You may use this license for any work of which you hold the copyright and which you wish to distribute. This license may be particularly suitable if your work is T_EX-related (such as a L^AT_EX package), but it is written in such a way that you can use it even if your work is unrelated to T_EX.

The section ‘WHETHER AND HOW TO DISTRIBUTE WORKS UNDER THIS LICENSE’, below, gives instructions, examples, and recommendations for authors who are considering distributing their works under this license.

This license gives conditions under which a work may be distributed and modified, as well as conditions under which modified versions of that work may be distributed.

We, the L^AT_EX3 Project, believe that the conditions below give you the freedom to make and distribute modified versions of your work that conform with whatever technical specifications you wish while maintaining the availability, integrity, and reliability of that work. If you do not see how to achieve your goal while meeting these conditions, then read the document ‘c_fgguide.tex’ and ‘modguide.tex’ in the base L^AT_EX distribution for suggestions.

Definitions

In this license document the following terms are used:

Work: Any work being distributed under this License.

Derived Work: Any work that under any applicable law is derived from the Work.

Modification: Any procedure that produces a Derived Work under any applicable law – for example, the production of a file containing an original file associated with the Work or a significant portion of such a file, either verbatim or with modifications and/or translated into another language.

Modify: To apply any procedure that produces a Derived Work under any applicable law.

Distribution: Making copies of the Work available from one person to another, in whole or in part. Distribution includes (but is not limited to) making any electronic components of the Work accessible by file transfer protocols such as FTP or HTTP or by shared file systems such as Sun’s Network File System (NFS).

Compiled Work: A version of the Work that has been processed into a form where it is directly usable on a

computer system. This processing may include using installation facilities provided by the Work, transformations of the Work, copying of components of the Work, or other activities. Note that modification of any installation facilities provided by the Work constitutes modification of the Work.

Current Maintainer: A person or persons nominated as such within the Work. If there is no such explicit nomination then it is the ‘Copyright Holder’ under any applicable law.

Base Interpreter: A program or process that is normally needed for running or interpreting a part or the whole of the Work.

A Base Interpreter may depend on external components but these are not considered part of the Base Interpreter provided that each external component clearly identifies itself whenever it is used interactively. Unless explicitly specified when applying the license to the Work, the only applicable Base Interpreter is a ‘L^AT_EX-Format’ or in the case of files belonging to the ‘L^AT_EX-format’ a program implementing the ‘T_EX language’.

Conditions on Distribution and Modification

- Activities other than distribution and/or modification of the Work are not covered by this license; they are outside its scope. In particular, the act of running the Work is not restricted and no requirements are made concerning any offers of support for the Work.
- You may distribute a complete, unmodified copy of the Work as you received it. Distribution of only part of the Work is considered modification of the Work, and no right to distribute such a Derived Work may be assumed under the terms of this clause.
- You may distribute a Compiled Work that has been generated from a complete, unmodified copy of the Work as distributed under Clause 2 above, as long as that Compiled Work is distributed in such a way that the recipients may install the Compiled Work on their system exactly as it would have been installed if they generated a Compiled Work directly from the Work.
- If you are the Current Maintainer of the Work, you

may, without restriction, modify the Work, thus creating a Derived Work. You may also distribute the Derived Work without restriction, including Compiled Works generated from the Derived Work. Derived Works distributed in this manner by the Current Maintainer are considered to be updated versions of the Work.

5. If you are not the Current Maintainer of the Work, you may modify your copy of the Work, thus creating a Derived Work based on the Work, and compile this Derived Work, thus creating a Compiled Work based on the Derived Work.
6. If you are not the Current Maintainer of the Work, you may distribute a Derived Work provided the following conditions are met for every component of the Work unless that component clearly states in the copyright notice that it is exempt from that condition. Only the Current Maintainer is allowed to add such statements of exemption to a component of the Work.
 - (a) If a component of this Derived Work can be a direct replacement for a component of the Work when that component is used with the Base Interpreter, then, wherever this component of the Work identifies itself to the user when used interactively with that Base Interpreter, the replacement component of this Derived Work clearly and unambiguously identifies itself as a modified version of this component to the user when used interactively with that Base Interpreter.
 - (b) Every component of the Derived Work contains prominent notices detailing the nature of the changes to that component, or a prominent reference to another file that is distributed as part of the Derived Work and that contains a complete and accurate log of the changes.
 - (c) No information in the Derived Work implies that any persons, including (but not limited to) the authors of the original version of the Work, provide any support, including (but not limited to) the reporting and handling of errors, to recipients of the Derived Work unless those persons have stated explicitly that they do provide such support for the Derived Work.
 - (d) You distribute at least one of the following with the Derived Work:
 - i. A complete, unmodified copy of the Work; if your distribution of a modified component is made by offering access to copy the modified component from a designated place, then offering equivalent access to copy the Work from the same or some similar place meets this condition, even though third parties are not compelled to copy the Work along with the modified component;
 - ii. Information that is sufficient to obtain a complete, unmodified copy of the Work.
7. If you are not the Current Maintainer of the Work, you may distribute a Compiled Work generated from a Derived Work, as long as the Derived Work is distributed to all recipients of the Compiled Work, and as long as the conditions of Clause 6, above, are met with regard to the Derived Work.
8. The conditions above are not intended to prohibit, and hence do not apply to, the modification, by any method, of any component so that it becomes identical to an updated version of that component of the Work as it is distributed by the Current Maintainer under Clause 4, above.
9. Distribution of the Work or any Derived Work in an alternative format, where the Work or that Derived Work (in whole or in part) is then produced by applying some process to that format, does not relax or nullify any sections of this license as they pertain to the results of applying that process.
10. (a) A Derived Work may be distributed under a different license provided that license itself honors the conditions listed in Clause 6 above, in regard to the Work, though it does not have to honor the rest of the conditions in this license.
 - (b) If a Derived Work is distributed under a different license, that Derived Work must provide sufficient documentation as part of itself to allow each recipient of that Derived Work to honor the restrictions in Clause 6 above, concerning changes from the Work.
11. This license places no restrictions on works that are unrelated to the Work, nor does this license place any restrictions on aggregating such works with the Work by any means.
12. Nothing in this license is intended to, or may be used to, prevent complete compliance by all parties with all applicable laws.

No Warranty

There is no warranty for the Work. Except when otherwise stated in writing, the Copyright Holder provides the Work ‘as is’, without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the Work is with you. Should the Work prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any author named in the components of the Work, or any other party who may distribute and/or modify the Work as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the Work or out of inability to use the Work (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by

anyone as a result of any failure of the Work to operate with any other programs), even if the Copyright Holder

or said author or said other party has been advised of the possibility of such damages.

Maintenance of The Work

The Work has the status ‘author-maintained’ if the Copyright Holder explicitly and prominently states near the primary copyright notice in the Work that the Work can only be maintained by the Copyright Holder or simply that it is ‘author-maintained’.

The Work has the status ‘maintained’ if there is a Current Maintainer who has indicated in the Work that they are willing to receive error reports for the Work (for example, by supplying a valid e-mail address). It is not required for the Current Maintainer to acknowledge or act upon these error reports.

The Work changes from status ‘maintained’ to ‘unmaintained’ if there is no Current Maintainer, or the person stated to be Current Maintainer of the work cannot be reached through the indicated means of communication for a period of six months, and there are no other significant signs of active maintenance.

You can become the Current Maintainer of the Work by agreement with any existing Current Maintainer to take over this role.

If the Work is unmaintained, you can become the Current Maintainer of the Work through the following steps:

1. Make a reasonable attempt to trace the Current Maintainer (and the Copyright Holder, if the two differ) through the means of an Internet or similar search.
2. If this search is successful, then enquire whether the Work is still maintained.
 - (a) If it is being maintained, then ask the Current Maintainer to update their communication data within one month.
 - (b) If the search is unsuccessful or no action to resume active maintenance is taken by the Current

Maintainer, then announce within the pertinent community your intention to take over maintenance. (If the Work is a L^AT_EX work, this could be done, for example, by posting to `comp.text.tex`.)

3. (a) If the Current Maintainer is reachable and agrees to pass maintenance of the Work to you, then this takes effect immediately upon announcement.
- (b) If the Current Maintainer is not reachable and the Copyright Holder agrees that maintenance of the Work be passed to you, then this takes effect immediately upon announcement.
4. If you make an ‘intention announcement’ as described in 2b above and after three months your intention is challenged neither by the Current Maintainer nor by the Copyright Holder nor by other people, then you may arrange for the Work to be changed so as to name you as the (new) Current Maintainer.
5. If the previously unreachable Current Maintainer becomes reachable once more within three months of a change completed under the terms of 3b or 4, then that Current Maintainer must become or remain the Current Maintainer upon request provided they then update their communication data within one month.

A change in the Current Maintainer does not, of itself, alter the fact that the Work is distributed under the LPPL license.

If you become the Current Maintainer of the Work, you should immediately provide, within the Work, a prominent and unambiguous statement of your status as Current Maintainer. You should also announce your new status to the same pertinent community as in 2b above.

Whether and How to Distribute Works under This License

This section contains important instructions, examples, and recommendations for authors who are considering distributing their works under this license. These authors are addressed as ‘you’ in this section.

Choosing This License or Another License

If for any part of your work you want or need to use *distribution* conditions that differ significantly from those in this license, then do not refer to this license anywhere in your work but, instead, distribute your work under a different license. You may use the text of this license as a model for your own license, but your license should not refer to the LPPL or otherwise give the impression that your work is distributed under the LPPL.

The document ‘`modguide.tex`’ in the base L^AT_EX distribution explains the motivation behind the conditions of this license. It explains, for example, why distributing

L^AT_EX under the GNU General Public License (GPL) was considered inappropriate. Even if your work is unrelated to L^AT_EX, the discussion in ‘`modguide.tex`’ may still be relevant, and authors intending to distribute their works under any license are encouraged to read it.

A Recommendation on Modification Without Distribution

It is wise never to modify a component of the Work, even for your own personal use, without also meeting the above conditions for distributing the modified component. While you might intend that such modifications will never be distributed, often this will happen by accident – you may forget that you have modified that component; or it may not occur to you when allowing others to access the modified version that you are thus distributing it and violating the conditions of this license in ways that could have

legal implications and, worse, cause problems for the community. It is therefore usually in your best interest to keep your copy of the Work identical with the public one. Many works provide ways to control the behavior of that work without altering any of its licensed components.

How to Use This License

To use this license, place in each of the components of your work both an explicit copyright notice including your name and the year the work was authored and/or last substantially modified. Include also a statement that the distribution and/or modification of that component is constrained by the conditions in this license.

Here is an example of such a notice and statement:

```
% pig.dtx
%% Copyright 2005 M. Y. Name
%
% This work may be distributed and/or modified under the
% conditions of the LaTeX Project Public License, either version 1.3
% of this license or (at your option) any later version.
% The latest version of this license is in
%   http://www.latex-project.org/lppl.txt
% and version 1.3 or later is part of all distributions of LaTeX
% version 2005/12/01 or later.
%
% This work has the LPPL maintenance status 'maintained'.
%
% The Current Maintainer of this work is M. Y. Name.
%
% This work consists of the files pig.dtx and pig.ins
% and the derived file pig.sty.
```

Given such a notice and statement in a file, the conditions given in this license document would apply, with the 'Work' referring to the three files 'pig.dtx', 'pig.ins', and 'pig.sty' (the last being generated from 'pig.dtx' using 'pig.ins'), the 'Base Interpreter' referring to any 'L^AT_EX-Format', and both 'Copyright Holder' and 'Current Maintainer' referring to the person 'M. Y. Name'.

If you do not want the Maintenance section of LPPL to apply to your Work, change 'maintained' above into 'author-maintained'. However, we recommend that you use 'maintained' as the Maintenance section was added in order to ensure that your Work remains useful to the community even when you can no longer maintain and support it yourself.

Derived Works That Are Not Replacements

Several clauses of the LPPL specify means to provide reliability and stability for the user community. They therefore concern themselves with the case that a Derived Work is intended to be used as a (compatible or incompatible) replacement of the original Work. If this is not the case (e.g., if a few lines of code are reused for a completely different task), then clauses 6b and 6d shall not apply.

Important Recommendations

Defining What Constitutes the Work

The LPPL requires that distributions of the Work contain all the files of the Work. It is therefore important that you provide a way for the licensee to determine which files constitute the Work. This could, for example, be achieved by explicitly listing all the files of the Work near the copyright notice of each file or by using a line such as:

```
% This work consists of all files listed in manifest.txt.
```

in that place. In the absence of an unequivocal list it might be impossible for the licensee to determine what is considered by you to comprise the Work and, in such a case, the licensee would be entitled to make reasonable conjectures as to which files comprise the Work.